

JZ4780

Mobile Application Processor

Programming Manual

Release Date: Apr. 10, 2015



北京君正集成电路股份有限公司
Ingenic Semiconductor Co.,Ltd.

JZ4780 Mobile Application Processor

Programming Manual

Copyright © 2005-2015 Ingenic Semiconductor Co. Ltd. All rights reserved.

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

Ingenic Semiconductor Co., Ltd.

**Ingenic Headquarters, East Bldg. 14, Courtyard #10,
Xibeiwang East Road, Haidian District, Beijing 100193, China**
Tel: 86-10-56345000
Fax: 86-10-56345001
Http: //www.ingenic.com

CONTENTS

CONTENTS	i
TABLES	i
FIGURES	i

Section 1 Overview

1 Overview	2
1.1 Block Diagram	3
1.2 Features	3
1.2.1 CPU	3
1.2.2 VPU	4
1.2.3 GPU	4
1.2.4 Display	5
1.2.5 Camera	7
1.2.6 Audio	7
1.2.7 Memory Interface	9
1.2.8 System	9
1.2.9 Peripheral	11
1.2.10 Bootrom	14
1.3 Characteristic	14

Section 2 Core Functions

2 CPU	16
2.1 Overview	16
2.2 CP0 Register Description	16
2.2.1 Cores Control (CP0 Register 12, Select 2)	16
2.2.2 Cores Status (CP0 Register 12, Select 3)	17
2.2.3 Reset Entry & IRQ Mask (CP0 Register 12, Select 4)	17
2.2.4 SPINLOCK (CP0 Register 12, Select 5)	18
2.2.5 SPINATOMIC (CP0 Register 12, Select 6)	18
2.2.6 Processor Identification (CP0 Register 15, Select 0)	19
2.2.7 EBase (CP0 Register 15, Select 1)	19
2.2.8 Config1 Register (CP0 Register 16, Select 1)	19
2.2.9 Config7 Register (CP0 Register 16, Select 7)	20
2.2.10 Mailbox0 (CP0 Register 20, Select 0)	21
2.2.11 Mailbox1 (CP0 Register 20, Select 1)	21
2.3 Application Notes	22

2.3.1	Cache Attribute	22
2.3.2	Mapping of IRQs observed by CPU core	23
2.3.3	EJTAG debug for multiple cores	23
3	VPU	24
3.1	Block Diagram	24
3.2	Features of VPU	25
3.3	Internal physical address base definition	26
3.4	AUX	27
3.4.1	Register Definition	27
3.5	TCSM/SRAM	30
3.5.1	TCSM/SRAM space usage	30
3.6	Video Acceleration Block	30
4	GPU	31
4.1	3D graphics	31
4.1.1	Introduction	31
4.1.2	PowerVR SGX Architecture - Key Features	31
4.1.3	SGX540 – 3D Features	31
4.1.4	SGX540 – Performance	33
4.2	EXtreme 2D	33
4.2.1	Overview	33
4.2.2	Registers Descriptions	34
4.3	Software stack	46

Section 3 Display/Camera/Audio

5	LCD Controller	48
5.1	Overview	48
5.2	Pin Description	49
5.3	Block Diagram	50
5.4	LCD Display Timing	52
5.5	OSD Graphic	53
5.5.1	Color Key	53
5.6	Register Description	55
5.6.1	Configure Register (LCDCFG)	57
5.6.2	Control Register (LCDCTRL)	59
5.6.3	Status Register (LCDSTATE)	60
5.6.4	OSD Configure Register (LCDOSDC)	60
5.6.5	OSD Control Register (LCDOSDCTRL)	62
5.6.6	OSD State Register (LCDOSDS)	62
5.6.7	Background0 Color Register (LCDBG0)	63
5.6.8	Background1 Color Register (LCDBG1)	63

5.6.9	Foreground Color Key Register 0 (LCDKEY0).....	63
5.6.10	Foreground Color Key Register 1 (LCDKEY1).....	64
5.6.11	ALPHA Register (LCDALPHA)	64
5.6.12	IPU Restart (LCDIPUR).....	65
5.6.13	RGB Control (LCDRGBC)	65
5.6.14	Virtual Area Setting (LCDVAT)	67
5.6.15	Display Area Horizontal Start/End Point (LCDDAH).....	67
5.6.16	Display Area Vertical Start/End Point (LCDDAV)	68
5.6.17	Foreground 0 XY Position Register (LCDXYP0)	68
5.6.18	Foreground 1 XY Position Register (LCDXYP1)	68
5.6.19	Foreground 0 Size Register (LCDSIZE0)	69
5.6.20	Foreground 1 Size Register (LCDSIZE1)	69
5.6.21	Vertical Synchronize Register (LCDVSYNC)	69
5.6.22	Horizontal Synchronize Register (LCDHSYNC).....	70
5.6.23	PS Signal Setting (LCDPS)	70
5.6.24	CLS Signal Setting (LCDCLS)	71
5.6.25	SPL Signal Setting (LCDSPL)	71
5.6.26	REV Signal Setting (LCDREV).....	71
5.6.27	Interrupt ID Register (LCDIID).....	72
5.6.28	Descriptor Address Registers (LCDDAx)	72
5.6.29	Source Address Registers (LCDSA)	73
5.6.30	Frame ID Registers (LCDFIDx)	73
5.6.31	DMA Command Registers (LCDCMDx)	74
5.6.32	DMA OFFSIZE Registers (LCDOFFSx)	75
5.6.33	DMA Page Width Registers (LCDPWx)	76
5.6.34	DMA Command Counter Registers (LCDCNUMx)	76
5.6.35	DMA Command Counter Registers (LCDCPOSx)	76
5.6.36	Foreground x Size in Descriptor (LCDESSIZEx).....	77
5.6.37	Priority level threshold configure Register (LCDPCFG).....	78
5.6.38	Dual LCD Channel Control(LCDCDUALCTRL).....	79
5.6.39	Image Enhancement CFG (LCDENH_CFG).....	79
5.6.40	Color Space Conversion CFG (LCDENH_CSCCFG)	80
5.6.41	LUMA CFG (LCDENH_LUMACFG)	81
5.6.42	CHROMA0 CFG (LCDENH_CHROCFG0)	82
5.6.43	CHROMA1 CFG (LCDENH_CHROCFG1)	82
5.6.44	DITHER CFG (LCDENH_DITHERCFG)	82
5.6.45	Enhance status (LCDENH_STATUS).....	83
5.6.46	GAMMA CFG (LCDENH_GAMMA)	83
5.6.47	VEE CFG (LCDENH_VEE)	84
5.7	LCD Controller Pin Mapping	84
5.7.1	TFT Pin Mapping	84
5.7.2	Data mapping to GPIO function.	86
5.8	Display Timing.....	86

5.8.1	General 16-bit and 18-bit TFT Timing.....	86
5.8.2	8-bit Serial TFT Timing.....	87
5.8.3	Special TFT Timing	88
5.8.4	Delta RGB panel timing	90
5.8.5	RGB Dummy mode timing.....	90
5.9	Format of Frame Buffer	91
5.9.1	16bpp	91
5.9.2	18bpp	91
5.9.3	24bpp	92
5.9.4	16bpp with alpha.....	92
5.9.5	18bpp with alpha.....	92
5.9.6	24bpp with alpha.....	92
5.9.7	24bpp compressed	92
5.10	Format of Data Pin Utilization.....	93
5.10.1	18-bit Parallel TFT	93
5.10.2	16-bit Parallel TFT	93
5.10.3	8-bit Serial TFT (24bpp).....	93
5.11	LCD Controller Operation.....	94
5.11.1	Set LCD Controller AHB Clock and Pixel Clock	94
5.11.2	Enabling the Controller	94
5.11.3	Disabling the Controller	94
5.11.4	Resetting the Controller	95
5.11.5	Frame Buffer	95
5.11.6	CCIR601/CCIR656	95
5.11.7	OSD Operation	95
5.11.8	Descriptor Operation.....	96
5.11.9	IPU direct connect mode	97
5.11.10	VGA output.....	97
6	Smart LCD Controller.....	98
6.1	Overview.....	98
6.2	Structure	98
6.3	Pin Description	99
6.4	Register Description	99
6.4.1	SLCD Configure Register (MCFG).....	100
6.4.2	SLCD Control Register (MCTRL)	101
6.4.3	SLCD Status Register (MSTATE)	102
6.4.4	SLCD Data Register (MDATA).....	102
6.5	System Memory Format.....	103
6.5.1	Data format	103
6.5.2	Command Format.....	103
6.6	Transfer Mode	104
6.6.1	DMA Transfer Mode.....	104

6.6.2	Register Transfer Mode.....	105
6.7	Timing	105
6.7.1	Parallel Timing.....	105
6.7.2	Serial Timing.....	105
6.8	Operation Guide.....	106
6.8.1	DMA Operation.....	106
6.8.2	Register Operation	107
7	Decompressor	108
7.1	Overview	108
7.2	Compress Method.....	108
7.3	Operation Guide.....	110
8	Image Enhancement	111
8.1	Overview	111
8.2	Operation Guide.....	111
9	Image Process Unit	112
9.1	Overview	112
9.1.1	Feature	112
9.2	Block	113
9.3	Data flow	113
9.3.1	Input data.....	113
9.3.2	Output data.....	113
9.3.3	Resize Coefficients LUT	113
9.4	Registers Descriptions	114
9.4.1	IPU global Control Register.....	115
9.4.2	IPU OSD Control Register.....	115
9.4.3	IPU Frame Control Register	116
9.4.4	IPU Status Register	117
9.4.5	IPU trigger register	118
9.4.6	IPU register configure enable control register.....	118
9.4.7	Data Format Register	119
9.4.8	Input Y Data Address Register	120
9.4.9	Input U Data Address Register.....	121
9.4.10	Input V Data Address Register	121
9.4.11	Input source TLB base address	122
9.4.12	Destination TLB base address	122
9.4.13	Input Y Data Address of next frame Register.....	123
9.4.14	Input U Data Address of next frame Register.....	123
9.4.15	Input V Data Address of next frame Register.....	123
9.4.16	Source TLB base address of next frame.....	124
9.4.17	Destination TLB base address of next frame	124

9.4.18	ADDRESS Mapping.....	125
9.4.19	Input Geometric Size Register.....	125
9.4.20	Input Y Data Line Stride Register	126
9.4.21	Input UV Data Line Stride Register.....	126
9.4.22	Output Frame Start Address Register.....	126
9.4.23	Output Data Address of next frame Register.....	127
9.4.24	Output Frame offset on screen Register	127
9.4.25	Output Geometric Size Register	128
9.4.26	Output Data Line Stride Register	128
9.4.27	CSC C0 Coefficient Register	129
9.4.28	CSC C1 Coefficient Register	129
9.4.29	CSC C2 Coefficient Register	130
9.4.30	CSC C3 Coefficient Register	130
9.4.31	CSC C4 Coefficient Register	130
9.4.32	Resize Coefficients Table Index Register	131
9.4.33	Horizontal Resize Coefficients Look Up Table Register group	131
9.4.34	Vertical Resize Coefficients Look Up Table Register group	136
9.4.35	Calculation for Resized width and height	137
9.4.36	CSC Offset Parameter Register	138
9.5	IPU Operation Flow	139
9.5.1	Data out to frame buffer	139
9.5.2	Data out to lcdc.....	140
9.5.3	Operation example	141
9.6	Special Instruction	143
9.6.1	Resizing size feature	143
9.6.2	Color convention feature.....	143
9.6.3	YUV/YCbCr to RGB CSC Equations.....	144
9.6.4	Output data package format (RGB order).....	144
9.6.5	Input data package format (RGB order)	145
9.6.6	Source Data storing format in external memory (separated YUV Frame).....	145
10	LVDS Controller	147
10.1	Overview.....	147
10.2	Register Description	147
10.2.1	TXCTRL (LVDS Transmitter Control Register)	147
10.2.2	TXPLL0 (LVDS Transmitter's PLL Control Register 0)	149
10.2.3	TXPLL1 (LVDS Transmitter's PLL Control Register 0)	150
10.2.4	TXECTRL (LVDS Transmitter's Enhance Control)	150
10.3	Operate Description	151
10.3.1	Operate Mode	151
10.3.2	LVDS Output Amplitude Control	151
10.3.3	LVDS Input Clock Edge-Delay Control	152
10.3.4	Output data start-edge Control	153

10.3.5	PLL Input Divider Value Setting	153
10.3.6	PLL Feedback Divider Value Setting	153
10.3.7	VCO Gain Calibration	154
10.3.8	PLL Output Clock Frequency Setting	154
10.3.9	LVDS-LCDC RGB Data Mapping	154
10.3.10	LVDS VESA/JEIDA Mode	155
10.4	Operate Flow	156
10.4.1	Power Down Mode	156
10.4.2	LVDS Output Mode	157
10.4.3	RGB CMOS Output Mode	157
11	Camera Interface Module	158
11.1	Overview	158
11.1.1	Features	158
11.1.2	Pin Description	158
11.2	CIM Special Register	159
11.2.1	CIM Register Map	159
11.2.2	CIM Configuration Register (CIMCFG)	159
11.2.3	CIM Control Register (CIMCR)	162
11.2.4	CIM Control Register 2 (CIMCR2)	163
11.2.5	CIM Status Register (CIMST).....	165
11.2.6	CIM Interrupt Mask Register (CIMIMR).....	167
11.2.7	CIM Interrupt ID Register (CIMIID).....	167
11.2.8	CIM Descriptor Address (CIMDA)	168
11.2.9	CIM Frame buffer Address Register (CIMFA)	168
11.2.10	CIM Frame ID Register (CIMFID).....	169
11.2.11	CIM DMA Command Register (CIMCMD).....	169
11.2.12	CIM Window Size (CIMWSIZE).....	170
11.2.13	CIM Window Offset (CIMWOFFSET).....	171
11.2.14	CIM Frame Size Register (CIMFS)	171
11.2.15	CIM Y Frame buffer Address Register (CIMYFA).....	172
11.2.16	CIM Y DMA Command Register (CIMYCMD)	172
11.2.17	CIM Cb Frame buffer Address Register (CIMCBFA).....	173
11.2.18	CIM Cb DMA Command Register (CIMCBCMD)	173
11.2.19	CIM Cr Frame buffer Address Register (CIMCRFA)	174
11.2.20	CIM Cr DMA Command Register (CIMCRCMD).....	174
11.2.21	CIM TLB Control Register (CIMTC)	175
11.2.22	CIM TLB Index Register (CIMTINX).....	175
11.2.23	CIM TLB Content Register (CIMTCNT).....	176
11.3	CIM Data Sampling Modes	176
11.3.1	Gated Clock Mode.....	176
11.3.2	ITU656 Interlace Mode.....	177
11.3.3	ITU656 Progressive Mode.....	178

11.4	DMA Descriptors.....	179
11.4.1	4-Word Descriptor.....	179
11.4.2	8-Word Descriptor.....	179
11.5	Interrupt Generation.....	180
11.6	Software Operation.....	181
11.6.1	Enable CIM with DMA.....	181
11.6.2	Enable CIM without DMA.....	181
11.6.3	Disable CIM.....	181
12	AC97/I2S/SPDIF Controller.....	182
12.1	Overview.....	182
12.1.1	Block Diagram.....	183
12.1.2	Features.....	183
12.1.3	Interface Diagram.....	184
12.1.4	Signal Descriptions.....	187
12.2	Register Descriptions.....	189
12.2.1	AIC Configuration Register (AICFR).....	191
12.2.2	AIC Common Control Register (AICCR).....	194
12.2.3	AIC AC-link Control Register 1 (ACCR1).....	197
12.2.4	AIC AC-link Control Register 2 (ACCR2).....	198
12.2.5	AIC I2S/MSB-justified Control Register (I2SCR).....	199
12.2.6	AIC Controller FIFO Status Register (AICSR).....	200
12.2.7	AIC AC-link Status Register (ACSR).....	202
12.2.8	AIC I2S/MSB-justified Status Register (I2SSR).....	203
12.2.9	AIC AC97 CODEC Command Address & Data Register (ACCAR, ACCDR).....	204
12.2.10	AIC AC97 CODEC Status Address & Data Register (ACSAR, ACSDR).....	205
12.2.11	AIC I2S/MSB-justified Clock Divider Register (I2SDIV).....	206
12.2.12	AIC FIFO Data Port Register (AICDR).....	206
12.2.13	SPDIF Enable Register (SPENA).....	207
12.2.14	SPDIF Control Register (SPCTRL).....	207
12.2.15	SPDIF State Register (SPSTATE).....	208
12.2.16	SPDIF Configure 1 Register (SPCFG1).....	209
12.2.17	SPDIF Configure 2 Register (SPCFG2).....	210
12.2.18	SPDIF FIFO Register (SPFIFO).....	212
12.3	Serial Interface Protocol.....	212
12.3.1	AC-link serial data format.....	212
12.3.2	I2S and MSB-justified serial audio format.....	213
12.3.3	Audio sample data placement in SDATA_IN/SDATA_OUT.....	215
12.3.4	SPDIF Protocol.....	217
12.4	AC97/I2S Operation.....	217
12.4.1	Initialization.....	218
12.4.2	AC '97 CODEC Power Down.....	219
12.4.3	Cold and Warm AC '97 CODEC Reset.....	219

12.4.4	External CODEC Registers Access Operation.....	220
12.4.5	Audio Replay	221
12.4.6	Audio Record.....	222
12.4.7	FIFOs operation	223
12.4.8	Data Flow Control.....	225
12.4.9	Audio Samples format.....	226
12.4.10	Serial Audio Clocks and Sampling Frequencies	228
12.4.11	Interrupts.....	232
12.5	SPDIF Guide.....	232
12.5.1	Set SPDIF clock frequency	232
12.5.2	PCM audio mode operation (Reference IEC60958)	232
12.5.3	Non-PCM mode operation (Reference IEC61937)	233
12.5.4	Disable operation	233
13	PCM Interface	234
13.1	Overview	234
13.2	Pin Description.....	234
13.3	Block Diagram.....	235
13.4	Register Description.....	235
13.4.1	PCM Control Register (PCMCTL)	235
13.4.2	PCM Configuration Register (PCMCFG)	237
13.4.3	PCM FIFO DATA PORT REGISTER (PCMDP)	238
13.4.4	PCM INTERRUPT CONTROL REGISTER (PCMINTC).....	238
13.4.5	PCM INTERRUPT STATUS REGISTER (PCMINTS).....	239
13.4.6	PCM CLOCK DIVIDE REGISTER (PCMDIV).....	240
13.5	PCM Interface Timing	241
13.5.1	Short Frame SYN	241
13.5.2	Long Frame SYN.....	242
13.5.3	Multi-Slot Operation.....	243
13.6	PCM Operation	243
13.6.1	PCM Initialization.....	243
13.6.2	Audio Replay	244
13.6.3	Audio Record.....	244
13.6.4	FIFOs operation	245
13.6.5	Data Flow Control.....	246
13.6.6	PCM Serial Clocks and Sampling Frequencies	246
13.6.7	Interrupts	247
14	Internal CODEC Interface.....	248
14.1	Overview	248
14.2	Features.....	248
14.2.1	Signal Descriptions.....	249
14.2.2	Block Diagram	250

14.2.3	Application schematic	251
14.3	Mapped Register Descriptions	252
14.3.1	CODEC internal register access control (RGADW).....	252
14.3.2	CODEC internal register data output (RGDATA)	253
14.4	Operation.....	254
14.4.1	Access to internal registers of the embedded CODEC	254
14.4.2	CODEC controlling and typical operations	254
14.4.3	Power saving	255
14.4.4	Pop noise and the reduction of it	255
14.5	Timing parameters.....	256
14.6	C parameters.....	257
14.7	CODEC internal Registers	258
14.7.1	CODEC internal registers	259
14.8	Programmable gains	286
14.8.1	Programmable boost gain: GIM.....	286
14.8.2	Programmable input gain amplifier: GID	286
14.8.3	Programmable digital attenuation: GOD.....	287
14.8.4	Programmable attenuation: GO.....	287
14.8.5	Programmable Bypass path attenuation: GI.....	288
14.8.6	Programmable digital mixer gain: GIMIX and GOMIX.....	289
14.8.7	Gain refresh strategy	289
14.9	Configuration of the headphone output stage	290
14.10	Out-of-band noise filtering	290
14.11	Output short-circuit protection (headphone output).....	290
14.11.1	Indication of the short circuit detection	290
14.11.2	Reset of short circuit detection.....	291
14.11.3	Capacitor-coupled headphone connection	291
14.12	Sampling frequency: FREQ.....	291
14.13	Programmable data word length	292
14.14	Ramping system note.....	292
14.15	AGC system guide	293
14.15.1	AGC operating mode	293
14.16	Digital Mixer description	295
14.17	Digital microphone interface.....	297
14.17.1	Timing Diagram.....	298
14.17.2	Timings.....	298
14.17.3	Noise template (TBC)	299
14.18	CODEC Operating modes.....	299
14.18.1	Power-On mode and Power-Off mode	300
14.18.2	RESET mode	300
14.18.3	STANDBY mode	301
14.18.4	SLEEP mode.....	301
14.18.5	Initial all the gain	302

14.18.6	Soft Mute mode	302
14.18.7	Power-Down mode and ACTIVE mode	303
14.18.8	Working modes summary	303
14.19	SYS_CLK turn-off and turn-on	304
14.20	Requirements on outputs and inputs selection and power-down modes	305
14.21	Anti-pop operation sequences	305
14.21.1	Initialization and configuration	305
14.21.2	Start up sequence (DAC).....	305
14.21.3	Shutdown sequence (DAC).....	307
14.21.4	Start up sequence (Line input)	308
14.21.5	Shutdown sequence (Line input).....	309
14.22	Circuits design suggestions	310
14.22.1	Avoid quiet ground common currents.....	310
14.22.2	Headphone connection (Capacitor-coupled).....	311
14.22.3	Microphone connection	311
14.22.4	Description of the connections to the jack.....	314
14.22.5	PCB considerations	315
14.23	Analog characteristics.....	316
14.23.1	Line input to audio ADC path.....	316
14.23.2	Audio DAC to headphone output path.....	317
14.23.3	Audio DAC to mono line output path	319
14.23.4	Line input to headphone output path (analog bypass)	319
14.23.5	Micbias and reference	321

Section 4 Memory Interface

15	DDR Controller	323
15.1	Overview	323
15.1.1	Supported DDR SDRAM Types	323
15.1.2	Block Diagram	323
15.2	Register Description.....	324
15.2.1	DSTATUS	326
15.2.2	DCFG	327
15.2.3	DCTRL.....	329
15.2.4	DLMR	332
15.2.5	DTIMING1,2,3,4,5,6 (DDR Timing Configure Register)	334
15.2.6	DREFCNT (DDR Auto-Refresh Counter)	337
15.2.7	DMMAP0,1 (DDR Memory Map Configure Register)	338
15.2.8	DDL P (DDR DFI low power handshake control register)	339
15.2.9	DREMAP1,2,3,4,5 (DDR Address Remapping Register 1,2,3,4,5)	340
15.2.10	WCMDCTRL1 (Performance wcmd reorder & grouping).....	341
15.2.11	RCMDCTRL0 (Performance rcmd request control)	342
15.2.12	RCMDCTRL1 (Performance rcmd request control)	344

15.2.13	WDATTHD0 (performance wcmd request control)	344
15.2.14	WDATTHD1 (performance wcmd request control)	345
15.2.15	IPO RTPRI (performance priority control).....	345
15.2.16	CHQOS0,1,2,3,4,5 (performance QoS control)	347
15.2.17	CPM_DRCG.....	348
15.3	Functional Description	348
15.3.1	DDRC and DDR2 Memory Initialization Sequence	348
15.4	Change Clock Frequency	349
15.4.1	Manually SELF-REFRESH Mode	349
15.4.2	CPM driven SELF-REFRESH Mode.....	349
15.4.3	DLL bypass mode	350
15.5	Data Endian.....	350
15.6	DDR Connection Diagrams	350
15.6.1	Connection to one 512Mb x16 DDR2 device	350
15.6.2	Connection to two 512Mb x16 DDR2 devices	351
16	External NAND Memory Controller.....	352
16.1	Overview.....	352
16.2	Pin Description	352
16.3	Physical Address Space Map	353
16.4	Static Memory Interface.....	356
16.4.1	Register Description	356
16.4.2	Example of Connection.....	361
16.4.3	Basic Interface	362
16.4.4	Burst ROM Interface	365
16.5	NAND Flash Interface	366
16.5.1	Register Description	366
16.5.2	NAND Flash Boot Loader	382
16.5.3	NAND Flash Operation	383
16.5.4	Example of Toggle NAND Flash Access.....	384
17	BCH Controller.....	386
17.1	Overview.....	386
17.2	Register Description	388
17.2.1	BCH Control Register (BHCR).....	388
17.2.2	BCH Control Set Register (BHCSR).....	389
17.2.3	BCH Control Clear Register (BHCCR)	390
17.2.4	BCH ENC/DEC Count Register (BHCNT)	390
17.2.5	BCH Data Register (BHDR).....	391
17.2.6	BCH Parity Register (BHPARn, n=0, 1, 2, 3, 4, 5...26, 27)	391
17.2.7	BCH Error Report Register (BHERRn, n=0,1,2,3,4,5,6,7,...,61,62,63).....	392
17.2.8	BCH Interrupt Status Register (BHINT)	393
17.2.9	BCH Interrupt Enable Set Register (BHINTES).....	395

17.2.10	BCH Interrupt Enable Clear Register (BHINTEC)	395
17.2.11	BCH Interrupt Enable Register (BHINTE)	396
17.2.12	BCH User TAG OUTPUT Register (BHTO).....	397
17.3	BCH Operation.....	397
17.3.1	Encoding Sequence	397
17.3.2	Decoding Sequence	399

Section 5 System Functions

18	Clock Reset and Power Controller.....	402
18.1	Overview	402
18.1.1	CGU Block Diagram	403
18.1.2	CGU Registers	405
18.1.3	PLL Operation	441
18.1.4	Main Clock Division Change Sequence	442
18.2	Power Manager.....	442
18.2.1	Low-Power Modes and Function.....	442
18.2.2	Register Description	443
18.2.3	Doze Mode	456
18.2.4	IDLE Mode	457
18.2.5	SLEEP Mode	457
18.2.6	Power Down Mode	457
18.3	Reset Control Module	458
18.3.1	Register Description	458
18.3.2	Power On Reset	459
18.3.3	WDT Reset.....	459
19	Timer/Counter Unit	460
19.1	Overview	460
19.2	Pin Description.....	460
19.3	Register Description.....	460
19.3.1	Timer Control Register (TCSR)	462
19.3.2	Timer Data FULL Register (TDFR).....	464
19.3.3	Timer Data HALF Register (TDHR).....	465
19.3.4	Timer Counter (TCNT).....	465
19.3.5	Timer Counter Enable Register (TER)	465
19.3.6	Timer Counter Enable Set Register (TESR)	466
19.3.7	Timer Counter Enable Clear Register (TECR)	467
19.3.8	Timer Flag Register (TFR)	468
19.3.9	Timer Flag Set Register (TFSR).....	469
19.3.10	Timer Flag Clear Register (TFCR)	470
19.3.11	Timer Mast Register (TMR)	471
19.3.12	Timer Mask Set Register (TMSR)	472

19.3.13	Timer Mask Clear Register (TMCR)	472
19.3.14	Timer Stop Register (TSR).....	473
19.3.15	Timer Stop Set Register (TSSR).....	474
19.3.16	Timer Stop Clear Register (TSCR)	475
19.3.17	Timer Status Register (TSTR).....	476
19.3.18	Timer Status Set Register (TSTSR).....	477
19.3.19	Timer Status Clear Register (TSTCR)	477
19.3.20	Timer control mode Register (TCUMOD)	478
19.3.21	Timer fifo write data (TFWD).....	479
19.3.22	Timer fifo state Register (TFIFOSR)	479
19.4	Operation.....	480
19.4.1	Basic Operation in TCU1 Mode	480
19.4.2	Disable and Shutdown Operation in TCU1 Mode.....	480
19.4.3	Basic Operation in TCU2 Mode	480
19.4.4	Disable and Shutdown Operation in TCU2 Mode.....	481
19.4.5	Read Counter in TCU2 Mode	481
19.4.6	Pulse Width Modulator (PWM)	482
19.4.7	Trackball Input Waveform Detect.....	482
19.4.8	Basic Operation in FIFO Mode 1	483
19.4.9	Basic Operation in FIFO Mode 2	483
20	Operating System Timer.....	485
20.1	Overview.....	485
20.2	Register Description	486
20.2.1	Operating System Control Register (OSTCSR)	486
20.2.2	Operating System Timer Data Register (OSTDR).....	487
20.2.3	Operating System Timer Counter (OSTCNTL)	488
20.2.4	Operating System Timer Counter high 32 bits buffer (OSTCNTLBUF)	488
20.3	Operation.....	488
20.3.1	Basic Operation	488
20.3.2	Disable and Shutdown Operation	489
21	Interrupt Controller	490
21.1	Overview.....	490
21.2	Register Description	490
21.2.1	Interrupt Controller Source Register (ICSR0).....	491
21.2.2	Interrupt Controller Source Register (ICSR1).....	491
21.2.3	Interrupt Controller Mask Register (ICMR0)	491
21.2.4	Interrupt Controller Mask Register (ICMR1)	492
21.2.5	Interrupt Controller Mask Set Register (ICMSR0)	492
21.2.6	Interrupt Controller Mask Set Register (ICMSR1)	493
21.2.7	Interrupt Controller Mask Clear Register (ICMCR0).....	493
21.2.8	Interrupt Controller Mask Clear Register (ICMCR1).....	493

21.2.9	Interrupt Controller Pending Register (ICPR0).....	494
21.2.10	Interrupt Controller Pending Register (ICPR1).....	494
21.2.11	Interrupt Source Register0 for PDMA (DSR0).....	495
21.2.12	Interrupt Mask Register0 for PDMA (DMR0).....	495
21.2.13	Interrupt Pending Register0 for PDMA (DPR0).....	495
21.2.14	Interrupt Source Register1 to PDMA (DSR1).....	496
21.2.15	Interrupt Mask Register1 for PDMA (DMR1).....	496
21.2.16	Interrupt Pending Register1 for PDMA (DPR1).....	497
21.3	Software Considerations.....	497
22	Watchdog Timer	499
22.1	Overview	499
22.2	Register Description.....	499
22.2.1	Watchdog Control Register (TCSR).....	499
22.2.2	Watchdog Enable Register (TCER).....	500
22.2.3	Watchdog Timer Data Register (TDR).....	501
22.2.4	Watchdog Timer Counter (TCNT).....	501
22.3	Watchdog Timer Function.....	501
23	PDMA Controller.....	503
23.1	Features.....	503
23.2	Block Diagram.....	503
23.3	Memory Mapped Register Descriptions.....	504
23.3.1	DMA Channel Registers.....	504
23.3.2	Global Control Registers.....	504
23.4	DMA Channel Register Definition.....	505
23.4.1	DMA Source Address (DSAn, n = 0 ~ 31).....	505
23.4.2	DMA Target Address (DTAn, n = 0 ~ 31).....	505
23.4.3	DMA Transfer Count (DTCn, n = 0 ~ 31).....	505
23.4.4	DMA Request Types (DRTn, n = 0 ~ 31).....	506
23.4.5	DMA Channel Control/Status (DCSn, n = 0 ~ 31).....	507
23.4.6	DMA Channel Command (DCMn, n = 0 ~ 31).....	508
23.4.7	DMA Descriptor Address (DDAn, n = 0 ~ 31).....	513
23.4.8	DMA Stride Difference (DSDn, n = 0 ~ 31).....	513
23.5	DMA Global Register Definition.....	513
23.5.1	DMA Control.....	513
23.5.2	DMA Interrupt Pending (DIRQP).....	515
23.5.3	DMA Doorbell (DDB).....	515
23.5.4	DMA Doorbell Set (DDS).....	515
23.5.5	DMA Channel Programmable (DMACP).....	516
23.5.6	DMA Soft IRQ Pending.....	516
23.5.7	DMA Soft IRQ Mask.....	517
23.5.8	DMA Channel IRQ Pending to MCU.....	517

23.5.9	DMA Channel IRQ to MCU Mask	517
23.5.10	Programmable Channel Bound With INTC_IRQ	518
23.5.11	Special Channel 0 and Channel 1	518
23.6	MCU	519
23.6.1	MCU Control & Status.....	519
23.6.2	MCU Normal MailBox	519
23.6.3	MCU Security MailBox.....	520
23.6.4	MCU Interrupt	520
23.6.5	Multiple Bank Tightly Coupled Sharing Memory.....	521
23.6.6	Security ROM & Security RAM.....	521
23.6.7	CP0 Registers of MCU	521
23.6.8	Normal Exceptions Accepted by MCU.....	522
23.6.9	How to Boot MCU Up	522
23.6.10	SecurityCall Accepted by MCU.....	522
23.6.11	Interruptable SMD Mode.....	523
23.7	DMA manipulation	524
23.7.1	Descriptor Transfer Mode	524
23.7.2	No-Descriptor Transfer Mode	527
23.8	DMA Requests.....	527
23.8.1	Auto Request	528
23.8.2	On-Chip Peripheral Request.....	528
23.9	How to Use Programmable DMA Channel.....	528
24	SAR A/D Controller	529
24.1	Overview.....	529
24.2	Register Description	530
24.2.1	ADC Enable Register (ADENA).....	530
24.2.2	ADC Configure Register (ADCFG)	531
24.2.3	ADC Control Register (ADCTRL)	533
24.2.4	ADC Status Register (ADSTATE)	534
24.2.5	ADC Same Point Time Register (ADSAME).....	535
24.2.6	ADC Wait Pen Down Time Register (ADWAIT).....	535
24.2.7	ADC Touch Screen Data Register (ADTCH)	535
24.2.8	ADC VBAT Data Register (ADVDAT)	538
24.2.9	ADC AUX Data Register (ADADAT)	538
24.2.10	ADC Clock Divide Register (ADCLK)	539
24.2.11	ADC Command Register (ADCMD).....	540
24.3	SAR A/D Controller Guide	541
24.3.1	Power Down Mode	541
24.3.2	A Sample Touch Screen Operation.....	541
24.3.3	SLEEP mode Sample Operation	542
24.3.4	VBAT Sample Operation.....	543
24.3.5	AUX Sample Operation	543

24.3.6	Disable Touch Screen	543
24.3.7	Multi-touch Operation	543
24.3.8	Use Software Command Operation	544
24.3.9	Use 5-wire touch panel Operation.....	544
24.3.10	Use External Touch Screen Controller Operation	544
24.3.11	Use TSC to support keypad	544
25	Real Time Clock	549
25.1	Overview	549
25.1.1	Features	549
25.1.2	Signal Descriptions.....	549
25.2	Register Description.....	550
25.2.1	RTC Control Register (RTCCR)	551
25.2.2	RTC Second Register (RTCSR).....	553
25.2.3	RTC Second Alarm Register (RTCSAR)	553
25.2.4	RTC Regulator Register (RTCGR).....	553
25.2.5	Hibernate Control Register (HCR)	554
25.2.6	HIBERNATE mode Wakeup Filter Counter Register (HWFCR).....	555
25.2.7	Hibernate Reset Counter Register (HRCCR)	555
25.2.8	HIBERNATE Wakeup Control Register (HWCR)	556
25.2.9	HIBERNATE Wakeup Status Register (HWRSR)	556
25.2.10	Hibernate Scratch Pattern Register (HSPR)	557
25.2.11	Write Enable Pattern Register (WENR)	558
25.2.12	CLK32K Pin control register (CKPCR)	559
25.2.13	OWI Pin control register (OWIPCR)	560
25.2.14	Power on control register (PWRONCR)	560
25.3	Time Regulation	561
25.4	Operate Mode	562
25.4.1	Normal Mode	562
25.4.2	HIBERNATE Mode	563
25.5	Clock select.....	563
26	EFUSE Slave Interface (EFUSE).....	565
26.1	Overview	565
26.2	Register Description.....	565
26.2.1	EFUSE Control Register (EFUCTRL)	566
26.2.2	EFUSE Configure Register (EFUCFG)	566
26.2.3	EFUSE Status Register (EFUSTATE)	567
26.2.4	EFUSE Data Register (EFUDATAn).....	568
26.3	Flow.....	570
26.3.1	Write EFUSE Flow	570
26.3.2	Write Security Key Flow	570
26.3.3	Read EFUSE Flow	570

26.3.4	Read Security Key/Random Number Flow	570
--------	--	-----

Section 6 Peripherals

27	TS Slave Interface (TSSI)	573
27.1	Overview.....	573
27.2	Pin Description	573
27.3	Register Description	573
27.3.1	TSSI Enable Register (TSENA).....	574
27.3.2	TSSI Configure Register (TSCFG)	575
27.3.3	TSSI Control Register (TSCTRL)	577
27.3.4	TSSI State Register (TSSTAT).....	578
27.3.5	TSSI FIFO Register (TSFIFO).....	578
27.3.6	TSSI PID Enable Register (TSPEN).....	578
27.3.7	TSSI Data Number Register (TSNUM).....	579
27.3.8	TSSI Data Trigger Register (TSDTR)	579
27.3.9	TSSI PID Filter Registers (TSPID0~15)	580
27.3.10	TSSI DMA Descriptor Address (TSDDA).....	580
27.3.11	TSSI DMA Target Address (TSDTA)	580
27.3.12	TSSI DMA Identifier (TSDID)	581
27.3.13	TSSI DMA Command (TSDCMD).....	581
27.3.14	TSST DMA Status (TSDST).....	582
27.3.15	TSSI Transfer Control Register (TSTC).....	583
27.4	TSSI Timing.....	584
27.5	TSSI Guide	584
27.5.1	TSSI Operation without PID Filtering Function.....	584
27.5.2	TSSI Operation with PID Filtering Function.....	585
28	General-Purpose I/O Ports.....	586
28.1	Overview.....	586
28.1.1	GPIO Port A Summary.....	586
28.1.2	GPIO Port B Summary	587
28.1.3	GPIO Port C Summary	588
28.1.4	GPIO Port D Summary	589
28.1.5	GPIO Port E Summary	590
28.1.6	GPIO Port F Summary.....	591
28.2	Registers Description	593
28.2.1	PORT PIN Level Registers (PxPIN)	597
28.2.2	PORT Interrupt Registers (PxINT).....	597
28.2.3	PORT Interrupt Set Registers (PxINTS)	598
28.2.4	PORT Interrupt Clear Registers (PxINTC).....	598
28.2.5	PORT Mask Registers (PxMSK).....	599
28.2.6	PORT Mask Set Registers (PxMSKS).....	599

28.2.7	PORT Mask Clear Registers (PxMSKC)	599
28.2.8	PORT PAT1/Direction Registers (PxPAT1)	600
28.2.9	PORT PAT1/Direction Set Registers (PxPAT1S)	600
28.2.10	PORT PAT1/Direction Clear Registers (PxPAT1C)	601
28.2.11	PORT PAT0/Data Registers (PxPAT0)	601
28.2.12	PORT PAT0/Data Set Registers (PxPAT0S)	602
28.2.13	PORT PAT0/Data Clear Registers (PxPAT0C)	602
28.2.14	PORT FLAG Registers (PxFLG)	603
28.2.15	PORT FLAG Clear Registers (PxFLGC)	603
28.2.16	PORT PULL Disable Registers (PxPE)	604
28.2.17	PORT PULL Set Registers (PxPES)	604
28.2.18	PORT PULL Clear Registers (PxPEC)	604
28.3	Program Guide	605
28.3.1	Port Function Guide	605
29	SMB Controller	606
29.1	Overview	606
29.1.1	Features	606
29.1.2	Pin Description	606
29.2	Registers	607
29.2.1	Registers Memory Map	607
29.2.2	Registers and Fields Description	608
29.3	Operating Flow	630
29.3.1	SMB Behavior	630
29.3.2	Master Mode Operation	631
29.3.3	Slave Mode Operation	633
29.3.4	Disabling SMB	636
30	Synchronous Serial Interface	637
30.1	Overview	637
30.2	Pin Description	637
30.3	Register Description	638
30.3.1	Register Mapping	638
30.3.2	SSI Data Register (SSIDR)	638
30.3.3	SSI Control Register0 (SSICR0)	639
30.3.4	SSI Control Register1 (SSICR1)	641
30.3.5	SSI Status Register (SSISR)	644
30.3.6	SSI Interval Time Control Register (SSIITR)	645
30.3.7	SSI Interval Character-per-frame Control Register (SSIICR)	646
30.3.8	SSI Clock Generator Register (SSIGR)	646
30.3.9	SSI Receive Counter Register (SSIRCNT)	647
30.4	Functional Description	647
30.5	Data Formats	647

30.5.1	Motorola's SPI Format Details	648
30.5.2	TI's SSP Format Details	652
30.5.3	National Microwire Format Details	653
30.6	Interrupt Operation	655
31	UART Interface	656
31.1	Overview	656
Features	656
31.1.1	Pin Description	656
31.2	Register Descriptions	656
31.2.1	Register Mapping	657
31.2.2	UART Receive Buffer Register (URBR)	657
31.2.3	UART Transmit Hold Register (UTHR)	658
31.2.4	UART Divisor Latch Low/High Register (UDLLR / UDLHR)	658
31.2.5	UART Interrupt Enable Register (UIER)	659
31.2.6	UART Interrupt Identification Register (UIIR)	659
31.2.7	UART FIFO Control Register (UFCR)	661
31.2.8	UART Line Control Register (ULCR)	662
31.2.9	UART Line Status Register (ULSR)	663
31.2.10	UART Modem Control Register (UMCR)	665
31.2.11	UART Modem Status Register (UMSR)	666
31.2.12	UART Scratchpad Register	666
31.2.13	Infrared Selection Register (ISR)	667
31.2.14	UART M Register (UMR)	668
31.2.15	UART Add Cycle Register (UACR)	668
31.2.16	UART RXFIFO Counter Register (URCR)	668
31.2.17	UART TXFIFO Counter Register (UTCR)	669
31.3	Operation	669
31.3.1	UART Configuration	669
31.3.2	Data Transmission	670
31.3.3	Data Reception	670
31.3.4	Receive Error Handling	670
31.3.5	Modem Transfer	671
31.3.6	DMA Transfer	671
31.3.7	Slow IrDA Asynchronous Interface	671
31.3.8	For any frequency clock to use the UART	671
32	MMC/SD CE-ATA Controller	674
32.1	Overview	674
32.2	Features	674
32.3	Pins Description	674
32.4	Block Diagram	675
32.5	MMC/SD Controller Signal I/O Description	675

32.6	Register Description.....	676
32.6.1	MSC Control Register (MSC_CTRL)	677
32.6.2	MSC Control 2 Register (MSC_CTRL2)	679
32.6.3	MSC Status Register (MSC_STAT).....	680
32.6.4	MSC Clock Rate Register (MSC_CLKRT)	682
32.6.5	MSC Command and Data Control Register (MSC_CMDAT)	682
32.6.6	MSC Response Time Out Register (MSC_RESTO)	685
32.6.7	MSC Read Time Out Register (MSC_RDTO)	686
32.6.8	MSC Block Size Register (MSC_BLKLEN).....	686
32.6.9	MSC Block Count Register (MSC_NOB)	686
32.6.10	MSC Successfully-transferred Blocks Count Register (MSC_SNOB)	687
32.6.11	MSC Interrupt Mask Register (MSC_IMASK)	687
32.6.12	MSC Interrupt Flag Register (MSC_IFLG)	689
32.6.13	MSC Command Index Register (MSC_CMD)	691
32.6.14	MSC Command Argument Register (MSC_ARG).....	692
32.6.15	MSC Response FIFO Register (MSC_RES).....	692
32.6.16	MSC Receive FIFO Port Register (MSC_RXFIFO)	692
32.6.17	MSC Transmit FIFO Port Register (MSC_TXFIFO)	693
32.6.18	MSC Low Power Mode Register (MSC_LPM)	693
32.6.19	MSC DMA Control Register (MSC_DMxAC).....	694
32.6.20	MSC DMA Descriptor Address Register (MSC_DMxANDA).....	695
32.6.21	MSC DMA Data Address Register (MSC_DMxADA)	695
32.6.22	MSC DMA Data Length Register (MSC_DMxALEN)	696
32.6.23	MSC DMA Command Register (MSC_DMxACMD)	696
32.6.24	MSC RTFIFO Data Counter Register(MSC_RTCNT)	697
32.7	MMC/SD Functional Description.....	697
32.7.1	MSC Reset	698
32.7.2	MSC Card Reset	698
32.7.3	Voltage Validation	698
32.7.4	Card Registry	698
32.7.5	Card Access	699
32.7.6	Protection Management	701
32.7.7	Card Status.....	705
32.7.8	SD Status.....	708
32.7.9	SDIO	709
32.7.10	Clock Control	710
32.7.11	Application Specified Command Handling	710
32.8	MMC/SD Controller Operation.....	711
32.8.1	Data FIFOs	711
32.8.2	DMA and Program I/O	712
32.8.3	Start and Stop clock.....	714
32.8.4	Software Reset.....	714
32.8.5	Voltage Validation and Card Registry.....	715

32.8.6	Single Data Block Write	717
32.8.7	Single Block Read	718
32.8.8	Multiple Block Write	718
32.8.9	Multiple Block Read	719
32.8.10	Stream Write (MMC)	720
32.8.11	Stream Read (MMC)	721
32.8.12	Erase, Select/Deselect and Stop	721
32.8.13	SDIO Suspend/Resume.....	722
32.8.14	SDIO ReadWait.....	722
32.8.15	Operation and Interrupt.....	722
33	OTG Controller.....	725
33.1	Overview.....	725
33.2	Pin Description	725
33.3	Control and Status Overview	725
33.4	CSR Memory Map	726
33.4.1	Register Maps	727
33.4.2	Global CSR Map	727
33.4.3	Host Mode CSR Map	728
33.4.4	Device Mode CSR Map	728
33.4.5	Data FIFO (DFIFO) Access Register Map	730
33.4.6	Interrupt Hierarchy	731
33.5	Register Descriptions	732
33.5.1	Application Access to the CSRs	732
33.5.2	Overview of Commonly Used Register Bits.....	733
33.5.3	Global Registers	738
33.5.4	Host Mode Registers	776
33.5.5	Device Mode Registers.....	794

Section 7 Boot

34	XBurst Boot ROM Specification.....	829
34.1	Boot Select	829
34.2	Boot Procedure.....	829
34.3	NAND Boot Specification.....	831
34.4	USB Boot Specification	834
34.5	MSC0 Boot Specification.....	839
34.6	eMMC Boot Specification	840
34.7	msc1 boot Specification.....	841
34.8	Spi boot Specification	841

TABLES

Table 3-1 VPU Features	25
Table 3-2 VPU Internal physical address base definition	26
Table 3-3 TCSM/SRAM space usage	30
Table 4-1 register list	34
Table 5-1 LCD Controller Pins Description	49
Table 5-2 LCD Controller Registers Description.....	55
Table 6-1 SLCD Pins Description	99
Table 9-1 register list	114
Table 9-2 no mapping mode	141
Table 9-3 mapping mode	142
Table 10-1 LVDS Register Description	147
Table 10-2 Operation Mode Description.....	151
Table 10-3 LVDS Output Amplitude Control	151
Table 10-4 LVDS Output Swing Level Control.....	152
Table 10-5 LVDS Output Level Function	152
Table 10-6 LVDS_TX Clock Edge Delay Control.....	152
Table 10-7 Output data start-edge Calibration (1x Clock Mode)	153
Table 10-8 PLL Input Divider Configuration.....	153
Table 10-9 PLL Feedback Divider Configuration.....	154
Table 10-10 PLL VCO gain calibration	154
Table 10-11 PLL Post Divider Configuration.....	154
Table 11-1 Camera Interface Pins Description	158
Table 11-2 CIM Registers	159
Table 11-3 The modes and the corresponding signals used	176
Table 12-1 AIC Pins Description	187
Table 12-2 AIC Registers Description.....	189
Table 12-3 AIC I2S mode configurations	193
Table 12-4 Sample data bit relate to SDATA_IN/SDATA_OUT bit.....	216
Table 12-5 Cold AC '97 CODEC Reset Timing parameters	220
Table 12-6 Warm AC '97 CODEC Reset Timing Parameters.....	220
Table 12-7 Audio Sampling rate, BIT_CLK and SYS_CLK frequencies.....	229
Table 12-8 BIT_CLK divider setting	230
Table 12-9 Approximate common multiple of SYS_CLK for all sample rates	230
Table 12-10 CPM/AIC clock divider setting for various sampling rate if PLL = 270.64MHz.....	231
Table 12-11 PLL parameters and audio sample errors for EXCLK=12MHz.....	231
Table 13-1 PCM Interface Pins Description.....	234
Table 13-2 PCM Registers Description	235
Table 14-1 CODEC signal IO pin description.....	249
Table 14-2 Internal CODEC Mapped Registers Description (AIC Registers)	252
Table 15-1 DDRC Register	324
Table 16-1 NEMC Pin Description	352

Table 16-2 Physical Address Space Map	355
Table 16-3 Default Configuration of NEMC Chip Select Signals	355
Table 16-4 Static Memory Interface Registers	356
Table 16-5 NAND Flash Interface Registers	366
Table 17-1 BCH Registers	386
Table 18-1 CGU Registers Configuration	405
Table 18-2 Power/Reset Management Controller Registers Configuration	443
Table 19-1 PWM Pins Description	460
Table 21-1 INTC Register	490
Table 23-1 DMA Channel Registers (n=0~31)	504
Table 23-2 Global Control Registers	504
Table 23-3 Transfer Request Types	506
Table 23-4 Available RDIL.....	509
Table 23-5 TCSM space.....	521
Table 23-6 Descriptor Structure	525
Table 24-1 SADC Pin Description	529
Table 24-2 SADC Register Description.....	530
Table 25-1 Registers for real time clock.....	550
Table 25-2 Registers for hibernating mode.....	550
Table 25-3 Clock select registers	563
Table 27-1 TSSI Pin Description	573
Table 27-2 TSSI Register Description.....	574
Table 28-1 GPIO Port A summary	586
Table 28-2 GPIO Port B summary	587
Table 28-3 GPIO Port C summary	588
Table 28-4 GPIO Port D summary	589
Table 28-5 GPIO Port E summary	590
Table 28-6 GPIO Port F summary.....	591
Table 28-7 GPIO Registers	593
Table 29-1 SMB Pin Description	606
Table 29-2 Registers Memory Map-Address Base	607
Table 29-3 Registers Memory Map-Address Offset.....	607
Table 30-1 SSI Controller Pins Description.....	637
Table 30-2 SSI Serial Port Registers	638
Table 30-3 SSI Interrupts	655
Table 31-1 UART Pins Description	656
Table 31-2 UART Registers Description	657
Table 31-3 UART Interrupt Identification Register Description	660
Table 32-1 Command Token Format.....	676
Table 32-2 MMC/SD Data Token Format	676
Table 32-3 MMC/SD Controller Registers Map.....	676
Table 32-4 Command Data Block Structure.....	702
Table 32-5 Card Status Description	705

Table 32-6 SD Status Structure	708
Table 32-7 How to stop multiple block write	719
Table 32-8 How to stop multiple block read.....	720
Table 32-9 The mapping between Commands and Steps	723
Table 33-1 OTG Pins Description	725
Table 33-2 Core Global CSR Map (000h-3FFh).....	727
Table 33-3 Host Mode CSR Map (400h-7FFh).....	728
Table 33-4 Device Mode CSR Map (800h-BFFh)	729
Table 33-5 Data FIFO (DFIFO) Access Register Map.....	730
Table 33-6 List of Commonly Used Register Bits	734
Table 33-7 Control and Status Register: GOTGCTL	738
Table 33-8 Interrupt Register: GOTGINT	742
Table 33-9 AHB Configuration Register: GAHBCFG.....	743
Table 33-10 USB Configuration Register: GUSBCFG	746
Table 33-11 Reset Register: GRSTCTL	752
Table 33-12 Interrupt Register: GINTSTS	755
Table 33-13 Interrupt Mask Register: GINTMSK.....	761
Table 33-14 Host Mode Receive Status Debug Read/Status Read and Pop Registers: GRXSTSR/GRXSTSP	764
Table 33-15 Device Mode Receive Status Debug Read/Status Read and Pop Registers: GRXSTSR/GRXSTSP	764
Table 33-16 Receive FIFO Size Register: GRXFSIZ	765
Table 33-17 Non-Periodic Transmit FIFO Size Register: GNPTXFSIZ (Host Mode and Device Shared FIFO Mode)	765
Table 33-18 Non-Periodic Transmit FIFO Size Register: GNPTXFSIZ (Device Dedicated FIFO Mode).....	766
Table 33-19 Non-Periodic Transmit FIFO/Queue Status Register: GNPTXSTS.....	766
Table 33-20 User HW Config1 Register: GHWCFG1.....	767
Table 33-21 User HW Config2 Register: GHWCFG2.....	768
Table 33-22 User HW Config3 Register: GHWCFG3.....	770
Table 33-23 User HW Config4 Register: GHWCFG4.....	772
Table 33-24 Global DFIFO Software Config Register: GDFIFOCFG	774
Table 33-25 Host Periodic Transmit FIFO Size Register: HPTXFSIZ	774
Table 33-26 Device Periodic Transmit FIFO-n Register: DPTXFSIZn	775
Table 33-27 Device In Endpoint Transmit FIFO Size Register: (DIEPTXFn)	776
Table 33-28 Host Configuration Register: HCFG	776
Table 33-29 Host Frame Interval Register: HFIR	779
Table 33-30 Host Frame Number/Frame Time Remaining Register: HFNUM.....	779
Table 33-31 Host Periodic Transmit FIFO/Queue Status Register: HPTXSTS	780
Table 33-32 Host All Channels Interrupt Register: HAINC	781
Table 33-33 Host All Channels Interrupt Mask Register: HAINC_MSK.....	781
Table 33-34 Host Port Control and Status Register: HPRT.....	782
Table 33-35 Host Channel-n Characteristics Register: HCCHARn.....	785

Table 33-36 Host Channel-n Split Control Register: HCSPLTn	787
Table 33-37 Host Channel-n Interrupt Register: HCINTn	787
Table 33-38 Host Channel-n Interrupt Mask Register: HCINTMSKn.....	789
Table 33-39 Host Channel-n Transfer Size Register: HCTSIZn	790
Table 33-40 Host Channel-n DMA Address Register: HCDMA n	792
Table 33-41 Host Channel-n DMA Buffer Address Register: HCDMABn	794
Table 33-42 Host Frame List Base Address Register: HFLBAddr	794
Table 33-43 Device Configuration Register: DCFG	795
Table 33-44 Device Control Register: DCTL	797
Table 33-45 Minimum Duration for Soft Disconnect	801
Table 33-46 Device Status Register: DSTS	801
Table 33-47 Device IN Endpoint Common Interrupt Mask Register: DIEPMSK.....	802
Table 33-48 Device OUT Endpoint Common Interrupt Mask Register: DOEPMSK.....	803
Table 33-49 Device All Endpoints Interrupt Register: DAIN T.....	804
Table 33-50 Device Endpoints Interrupt Mask Register: DAIN TMSK.....	804
Table 33-51 Device IN Token Sequence Learning Queue Read Register 1: DTKNQR1	805
Table 33-52 Device IN Token Sequence Learning Queue Register 2: DTKNQR2.....	805
Table 33-53 Device IN Token Sequence Learning Queue Register 3: DTKNQR3.....	806
Table 33-54 Device IN Token Sequence Learning Queue Register 4: DTKNQR4.....	806
Table 33-55 Device VBUS Discharge Time Register: DVBUSDIS	806
Table 33-56 Device VBUS Pulsing Time Register (DVBUSPULSE)	807
Table 33-57 Device Threshold Control Register (DTHRCTL).....	807
Table 33-58 Device IN Endpoint FIFO Empty Interrupt Mask Register: DIEPEMPMSK	809
Table 33-59 Device Each Endpoint Interrupt Register: DEACHINT	809
Table 33-60 Device Each Endpoint Interrupt Register Mask: DEACHINTMSK.....	810
Table 33-61 Device Each In Endpoint-n Interrupt Register: DIEPEACHMSKn	810
Table 33-62 Device Each Out Endpoint-n Interrupt Register: DOEPEACHMSKn	811
Table 33-63 Device Control IN Endpoint 0 Control Register: DIEPCTL0	812
Table 33-64 Device OUT Endpoint 0 Control Register: DOEPCTL0	814
Table 33-65 Device Endpoint-n Control Register: DIEPCTLn/DOEPCTLn	815
Table 33-66 Device Endpoint-n Interrupt Register: DIEPINTn/DOEPINTn	819
Table 33-67 Device IN Endpoint 0 Transfer Size Register: DIEPTSIZ0	823
Table 33-68 Device OUT Endpoint 0 Transfer Size Register: DOEPTSIZ0	823
Table 33-69 Device Endpoint-n Transfer Size Register: DIEPTSIZn/DOEPTSIZn	824
Table 33-70 Device Endpoint-n DMA Address Register: DIEPDMA n/DOEPDMA n	825
Table 33-71 Device Endpoint-n DMA Buffer Address Register: DIEPDMABn/DOEPDMABn	826
Table 33-72 Device IN Endpoint Transmit FIFO Status Register: DTXFSTS n	826
Table 34-1 Boot Configuration of JZ4780	829
Table 34-2 The definition of 5 flags in NAND flash	831
Table 34-3 Transfer Types Used by the Boot Program.....	835
Table 34-4 Vendor Request 0 Setup Command Data Structure	837
Table 34-5 Vendor Request 1 Setup Command Data Structure	838
Table 34-6 Vendor Request 2 Setup Command Data Structure	838

Table 34-7 Vendor Request 3 Setup Command Data Structure.....	838
Table 34-8 Vendor Request 4 Setup Command Data Structure.....	838
Table 34-9 Vendor Request 5 Setup Command Data Structure.....	839
Table 34-10 SPI nor flash boot flag informations.....	842

FIGURES

Figure 1-1 JZ4780 Diagram.....	3
Figure 3-1 VPU Block Diagram	24
Figure 5-1 Block Diagram when use OSD mode	50
Figure 5-2 Block Diagram of TFT mode (not use OSD)	51
Figure 5-3 Block Diagram of HDMI interface.....	51
Figure 5-4 Display Parameters.....	52
Figure 5-5 OSD Graphic.....	53
Figure 5-6 General 16-bit and 18-bit TFT LCD Timing.....	87
Figure 5-7 8-bit serial TFT LCD Timing (24bpp).....	88
Figure 5-8 Special TFT LCD Timing 1	89
Figure 5-9 Special TFT LCD Timing 2	89
Figure 5-10 Delta RGB timing	90
Figure 5-11 RGB Dummy timing.....	91
Figure 9-1 The Block about the IPUData flow	113
Figure 10-1 VESA Mode LVDS output timing.....	156
Figure 10-2 JEIDA Mode LVDS output timing	156
Figure 11-1 Typical BT.656 Vertical Blanking Intervals for 625/50 Video Systems	178
Figure 11-2 ITU656 Progressive Mode	179
Figure 12-1 AIC Block Diagram	183
Figure 12-2 Interface to an External AC'97 CODEC Diagram	184
Figure 12-3 Interface to an External Master Mode I2S/MSB-Justified CODEC Diagram	185
Figure 12-4 Interface to an External Master Mode I2S CODEC Diagram	185
Figure 12-5 Interface to an External Slave Mode I2S/MSB-Justified CODEC Diagram	185
Figure 12-6 Interface to an External Slave Mode I2S CODEC Diagram	186
Figure 12-7 Interface to a HDMI Transmitter via I2S Diagram	186
Figure 12-8 Interface to a HDMI Transmitter via SPDIF Diagram.....	186
Figure 12-9 Interface to an internal Master Mode I2S CODEC Diagram.....	187
Figure 12-10 AC-link audio frame format	212
Figure 12-11 AC-link tag phase, slot 0 format	213
Figure 12-12 AC-link data phases, slot 1 ~ slot 12 format	213
Figure 12-13 I2S data format (A: LR mode)	213
Figure 12-14 I2S data format (B: RL mode)	214
Figure 12-15 MSB-justified data format (C: LR mode).....	214
Figure 12-16 MSB-justified data format (D: RL mode).....	215
Figure 12-17 Block format	217
Figure 12-18 Sub-frame format in PCM mode	217
Figure 12-19 Sub-frame format in non-PCM mode.....	217
Figure 12-20 Cold AC '97 CODEC Reset Timing	219
Figure 12-21 Warm AC '97 CODEC Reset Timing.....	220
Figure 12-22 Transmitting/Receiving FIFO access via APB Bus	224
Figure 12-23 One channel (Left) and Two channels (right) mode (16 bits packed mode)	226

Figure 12-24 Four channels (Left) and Six channels (right) mode (16 bits packed mode).....	226
Figure 12-25 Eight channels mode (16 bits packed mode)	227
Figure 12-26 One channel (Left) and Two channels (right) mode	227
Figure 12-27 Four channels (Left) and Six channels (right) mode	227
Figure 12-28 Eight channels mode	228
Figure 12-29 SYS_CLK, BIT_CLK and SYNC generation scheme	229
Figure 13-1 Short Frame SYN Timing (Shown with 16bit Sample)	241
Figure 13-2 Short Frame SYN Timing (Shown with 16bit Sample)	242
Figure 13-3 Long Frame SYN Timing (Shown with 16bit Sample)	242
Figure 13-4 Long Frame SYN Timing (Shown with 16bit Sample)	242
Figure 13-5 Multi-Slot Frame SYN Timing (Shown with two Slots and 8bit Sample)	243
Figure 13-6 Transmitting/Receiving FIFO access via APB Bus.....	245
Figure 13-7 PCMCLK and PCMSYN generation scheme	247
Figure 14-1 CODEC block diagram	250
Figure 14-2 Internal CODEC works with AIC	250
Figure 14-3 AGC adjusting waves	294
Figure 14-4 AGC adjust areas	294
Figure 14-5 Digital microphone interface connection	297
Figure 14-6 Digital microphone timing diagram at MCLK = 12 MHz	298
Figure 14-7 Digital microphone modulation noise reference spectrum	299
Figure 14-8 CODEC Power Diagram.....	300
Figure 14-9 DAC Gain up and gain down sequence	302
Figure 14-10 ADC Gain up and gain down sequence	303
Figure 14-11 Start up sequence.....	306
Figure 14-12 Shutdown sequence	307
Figure 14-13 Capacitor-coupled connection	311
Figure 14-14 Ground distributing	314
Figure 14-15 the bottom corner of chip PCB Layer	315
Figure 15-1 DDRC block diagram.....	324
Figure 16-1 Physical Address Space Map	354
Figure 16-2 Example of 16-Bit Data Width SRAM Connection.....	361
Figure 16-3 Example of 8-Bit Data Width SRAM Connection.....	362
Figure 16-4 Basic Timing of Normal Memory Read.....	363
Figure 16-5 Basic Timing of Normal Memory Write	363
Figure 16-6 Normal Memory Read Timing With Wait (Software Wait Only)	364
Figure 16-7 Normal Memory Write Timing With Wait (Software Wait Only)	364
Figure 16-8 Normal Memory Read Timing With Wait (Wait Cycle Insertion by WAIT# pin)	365
Figure 16-9 Burst ROM Read Timing (Software Wait Only)	366
Figure 16-10 Toggle NAND Page Write/Read Operation	370
Figure 16-11 Example of DQS Delay Line Auto-Detect-Operation.....	378
Figure 16-12 Basic Timing of Toggle NAND Write.....	380
Figure 16-13 Basic Timing of Toggle NAND Read.....	381
Figure 16-14 Basic Timing of Toggle NAND Page Write	381

Figure 16-15 Basic Timing of Toggle NAND Page Read.....	382
Figure 16-16 Structure of NAND Flash Boot Loader.....	382
Figure 16-17 Static Bank 1 Partition When NAND Flash is Used (an example).....	383
Figure 16-18 Example of 8-bit NAND Flash Connection.....	384
Figure 16-19 Toggle NAND Page Read Operation.....	384
Figure 16-20 Toggle NAND Page Program Operation.....	384
Figure 16-21 Toggle NAND Page Program Operation (Basic).....	385
Figure 16-22 Program Operation with Random Data Input.....	385
Figure 17-1 Block Diagram for BCH encoding.....	398
Figure 17-2 Block Diagram for BCH decoding.....	399
Figure 17-3 BCH decoding data flow.....	400
Figure 18-1 Block Diagram of PLL.....	441
Figure 23-1 Block Diagram of PDMA.....	503
Figure 23-2 Descriptor Transfer Flow.....	526
Figure 23-3 Example for Stride Transfer Mode.....	527
Figure 24-1 6x5 keypad circuit.....	545
Figure 24-2 Wait for pen-down (C=1100) circuit.....	546
Figure 24-3 Measure X-position (C=0010) circuit.....	547
Figure 24-4 Measure Y-position (C=0011) circuit.....	547
Figure 25-1 Core Power On directly.....	562
Figure 25-2 Core Power On-Off-On.....	562
Figure 25-3 RTC clock selection path.....	564
Figure 27-1 Timing waveform in parallel mode.....	584
Figure 27-2 Timing waveform in serial mode.....	584
Figure 30-1 SPI Single Character Transfer Format (PHA = 0).....	649
Figure 30-2 SPI Single Character Transfer Format (PHA = 1).....	649
Figure 30-3 SPI Back-to-Back Transfer Format.....	650
Figure 30-4 SPI Frame Interval Mode Transfer Format (ITFRM = 0, LFST = 0).....	651
Figure 30-5 SPI Frame Interval Mode Transfer Format (ITFRM = 1, LFST = 1).....	652
Figure 30-6 TI's SSP Single Transfer Format.....	652
Figure 30-7 TI's SSP Back-to-back Transfer Format.....	653
Figure 30-8 National Microwire Format 1 Single Transfer.....	654
Figure 30-9 National Microwire Format 1 Back-to-back Transfer.....	654
Figure 30-10 National Microwire Format 2 Read Timing.....	654
Figure 30-11 National Microwire Format 2 Write Timing.....	655
Figure 32-1 MMC/SD Controller Block Diagram.....	675
Figure 33-1 OTG CSR Memory Map.....	726
Figure 33-2 Interrupt Hierarchy.....	731
Figure 33-3 Core Interrupt Handler.....	732
Figure 34-1 Boot sequence diagram of JZ4780.....	830
Figure 34-2 the distribution and structure of the boot code in NAND.....	832
Figure 34-3 JZ4780 NAND Boot Procedure.....	834
Figure 34-4 USB Communication Flow.....	835

Figure 34-5 Typical Procedure of USB Boot	837
Figure 34-6 JZ4780 eMMC Boot Procedure	841
Figure 34-7 JZ4780 SPI Boot Procedure	843

Section 1

OVERVIEW

1 Overview

JZ4780 is a mobile application processor targeting for multimedia rich and mobile devices like tablet computer, smart phone, mobile digital TV, and GPS. This SOC introduces a kind of innovative architecture to fulfill both high performance mobile computing and high quality video decoding requirements addressed by mobile multimedia devices. JZ4780 provides high-speed CPU computing power, good 3D experience and fluent 1080p video replay.

The CPU (Central Processing Unit) core, equipped with 32kB instruction and 32kB data level 1 cache, and 512kB level 2 cache, operating at 1.2GHz, and full feature MMU function performs OS related tasks. At the heart of the CPU core is XBurst[®] processor engine. XBurst[®] is an industry leading microprocessor core which delivers superior high performance and best-in-class low power consumption. A hardware floating-point unit which compatible with IEEE754 is also included.

The VPU (Video Processing Unit) core is powered with another XBurst[®] processor engine. The SIMD instruction set implemented by XBurst[®] engine, in together with the on chip video accelerating engine and post processing unit, delivers high video performance. The maximum resolution of 1080p in the formats of H.264, VC-1, MPEG-2, MPEC-4, RealVideo and VP8 are supported in decoding, the maximum resolution of 1080p in the format of H.264 are supported in encoding.

The GPU (Graph Processing Unit) core supports numerous 2D/3D graphics applications. It delivers hardware acceleration for 2D and 3D graphics displays, and supports screen sizes range from the smallest cell phones to full HD 1080p displays. It supports the standard APIs such as OpenGL ES2.0 and 1.1, and Open VG. The OS of Android, Linux and Windows are supported. The GPU provides high performance, high quality graphics and low power consumption.

The memory interface supports a variety of memory types that allow flexible design requirements, including glueless connection to SLC NAND flash memory or up to 64-bit ECC MLC/TLC NAND flash memory and toggle NAND flash for cost sensitive applications. It provides the interface to DDR2, DDR3, LPDDR and LPDDR2 memory chips with lower power consumption.

On-chip modules such as audio CODEC, multi-channel SAR-ADC, AC97/I2S controller and camera interface offer designers a rich suite of peripherals for multimedia application. GPS baseband is embedded. The LCD controller support regular RGB, LVDS and HDMI transmitter, up to 2-channel 1920x1080 output(One channel must be LVDS or HDMI), WLAN, Bluetooth and expansion options are supported through high-speed SPI and MMC/SD/SDIO host controllers. The TS (Transport stream) interface provides enough bandwidth to connect to an external mobile digital TV demodulator. Other peripherals such as USB OTG and USB 2.0 host, UART and SPI as well as general system resources provide enough computing and connectivity capability for many applications.

1.1 Block Diagram

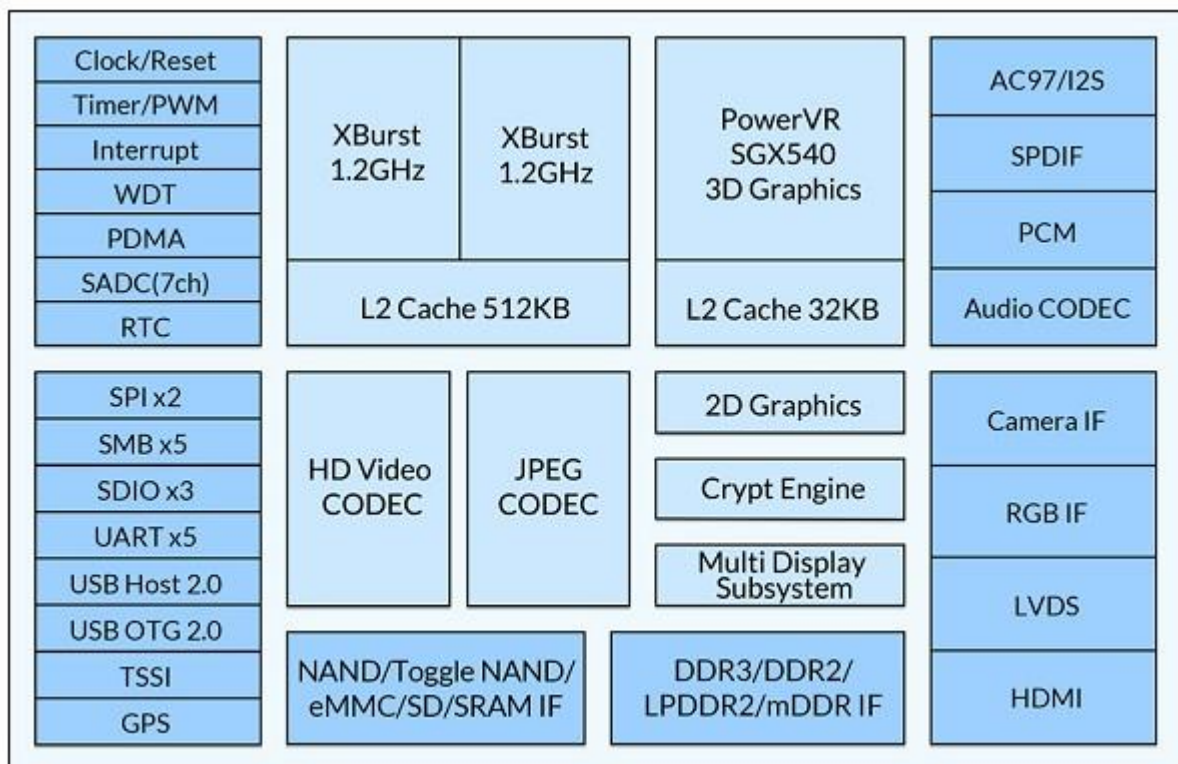


Figure 1-1 JZ4780 Diagram

1.2 Features

1.2.1 CPU

A symmetric XBurst[®]-1 dual-core implementation, with IRQ dispatching for each core and MSI protocol for cache coherence.

- XBurst[®]-1 core
 - XBurst[®] RISC instruction set
 - XBurst[®] SIMD instruction set
 - XBurst[®] FPU instruction set supporting both single and double floating point format which are IEEE754 compatible
 - XBurst[®] 9-stage pipeline micro-architecture, the operating frequency is 1.2G.
 - MMU
 - 32-entry joint-TLB
 - 8 entry Instruction TLB
 - 8 entry data TLB
 - L1 Cache
 - 32kB instruction cache

- 32kB data cache
- Hardware debug support
- 16kB tight coupled memory
- L2 Cache
 - 512kB unify cache
- CoreScheduler
 - Dedicated IRQ dispatcher managing IRQs to each core
- GDIR
 - Global directory aiding MSI protocol for cache coherence

1.2.2 VPU

Include Video Codec and JPEG Codec,

- MPEG-2 decoding up to 1080P 60fps
- VC-1 decoding up to 1080P 60fps
- H.264 decoding up to 1080P 60fps
- VP8 decoding up to 1080P 60fps
- MPEG-4 decoding up to 1080P 30fps
- RV9 decoding up to 1080P 30fps
- H.264 encoding up to 1080P 30fps
- JPEG compressing up to 100Mega-pixels per second (baseline)

1.2.3 GPU

- 3D graphics (PowerVR SGX540)
 - Deferred Pixel Shading
 - 32bit floating point depth accuracy with on chip depth buffer
 - 8-bit Stencil with on chip tile stencil buffer
 - 8 parallel depth/stencil tests per clock
 - Scissor test
 - Texture support
 - Cube Map
 - Projected Textures
 - Non square Textures
 - Texture Formats
 - RGBA 8888,565,1555,1565
 - Mono chromatic 8, 16, 16f, 32f, 32int
 - Dual channel, 8:8, 16:16, 16f:16f
 - Compressed Textures PVR-TC1, PVR-TC2, ETC1
 - Programmable support for YUV formats
 - Resolution Support
 - Frame buffer max size = 2048 x 2048
 - Texture max size = 2048 x 2048

- Texture Filtering
 - Bilinear, Trilinear, Anisotropic
 - Independent min and mag control
- Anti-aliasing
 - 4x Multisampling
 - up to 16x Full scene anti-aliasing
 - Programmable sample positions
- Indexed Primitive List support
 - Bus mastered
- Programmable vertex DMA
- Render to texture
 - Including twiddled formats
 - Auto MipMap generation
- 2D Graphics (X2D)
 - Location: AHB bus
 - Input format
 - Separate frame: YUV /YCbCr (4:2:0)
 - Packaged data: RGB888, RGB565, RGB555, NV12, NV21, TileYUV
 - Output data format
 - ARGB888, XRGB888, RGB555, RGB565
 - Color convention coefficient: configurable (CSC enable)
 - Minimum input image size (pixel): 4x4
 - Maximum input image size (pixel): 12288x12288 (12k x 12k)
 - Maximum output image size (pixel)
 - Width : up to 12288
 - Height: up to 12288
 - Image resizing
 - bi-cube zooming mode
 - Image Clockwise 90, 180, 270 rotation
 - Image horizontal and vertical mirror , same time with rotation
 - 5 layers OSD

1.2.4 Display

Include Multi-display subsystem and multiple display interface.

- LCD controller
 - Dual controller
 - Support dual panel(HDMI & TFT, or HDMI & LVDS, or TFT & LVDS, or HDMI & SLCD, or SLCD & LVDS)
 - Display size up to 4kx2k@30Hz(BPP24) for single panel, 1920x1080@60Hz(BPP24) for dual panel
 - Colors Supports
 - Encoded pixel data of 16, 18 or 24 BPP in TFT mode

- Support up to 16,777,216 (16M) colors in TFT mode
- Support 24/16 BPP compressed data
- Support 24 BPP packed data
- Panel Supports
 - Support 16-bit parallel TFT panel
 - Support 18-bit parallel TFT panel
 - Support 24-bit serial TFT panel with 8 data output pins
 - Support 24-bit parallel TFT panel
 - Support Delta RGB panel
 - Support SLCD panel
 - Support HDMI 1.4a Interface
 - Support LVDS Interface
- OSD Supports
 - Supports one single color background
 - Supports two foregrounds, and every size can be set for each foreground
 - Supports one transparency for the whole graphic
 - Supports one transparency for each pixel in one graphic
 - Supports color key and mask color key
 - Supports porter-duff blending
- Image Enhancement
 - Color space conversion: RGB to YCbCr, YCbCr to RGB
 - Support Contrast, Brightness, Hue, Saturation control
 - Support Visibility Enhance
 - Support Dither
 - Support Gamma Correction
- Image post processor(IPU)
 - Input format
 - Separate frame: YUV /YCbCr (4:2:0, 4:2:2, 4:4:4, 4:1:1), RGB888
 - Packaged data: YUV422, RGB888, RGB565, RGB555, YUV444
 - Separate frame in block format: YUV/YCbCr 420
 - Output data format
 - RGB (565, 555, 888, AAA)
 - Packaged data YUV422
 - Color convention coefficient: configurable (CSC enable)
 - Minimum input image size (pixel): 4x4
 - Maximum input image size (pixel): 8096x8096
 - Maximum output image size (pixel)
 - Width: up to 4095 (without vertical resizing)
 - up to 2048 (with vertical resizing)
 - Height: up to 4095
 - Image resizing
 - Support bilinear
 - 0 and bi-cube zooming mode

- Up scaling ratios up to 1:31 in fractional steps with 1/32 accuracy
- Down scaling ratios up to 31:1 in fractional steps with 1/32 accuracy

1.2.5 Camera

- Camera interface module
 - Input image size up to 4096x4096 pixels
 - Max. VGA for image preview
 - Max. VGA for video record
 - Integrated DMA
 - Supported data format: YCbCr 4:4:4, YCbCr 4:2:2 and other formats
 - Output format: csc mode is YCbCr 4:2:2 or YCbCr 4:2:0, bypass mode is the input data format
 - Output frame format
 - Packaged : for all data format
 - Separated: for YCbCr 4:4:4, YCbCr 4:2:2 and YCbCr 4:2:0
 - Supports ITU656 (YCbCr 4:2:2) input
 - Configurable CIM_VSYNC and CIM_HSYNC signals: active high/low
 - Configurable CIM_PCLK: active edge rising/falling
 - 256x33 image data receive FIFO (RXFIFO)
 - PCLK max. 80MHz
 - Configurable output order

1.2.6 Audio

- AC97/I2S/SPDIF controller
 - AC-link (AC97) features
 - Up to 20 bit audio sample data sizes supported
 - DMA transfer mode supported
 - Stop serial clock supported
 - Programmable Interrupt function supported
 - Support mono PCM data to stereo PCM data expansion on audio play back
 - Support endian switch on 16-bits normal audio samples play back
 - Support variable sample rate in AC-link format
 - Multiple channel output and double rated supported for AC-link format
 - Power Down Mode and two Wake-Up modes Supported for AC-link format
 - I2S features
 - 8, 16, 18, 20 and 24 bit audio sample data sizes supported, 16 bits packed sample data is supported
 - Up to 8 channels sample data supported
 - DMA transfer mode supported
 - Stop serial clock supported
 - Programmable Interrupt function supported

- Support share clock mode and split clock mode.
- Support mono PCM data to stereo PCM data expansion on audio play back
- Support endian switch on 16-bits normal audio samples play back
- Internal programmable or external serial clock and optional system clock supported for I2S or MSB-Justified format
- Internal I2S CODEC supported
- Two FIFOs for transmit and receive respectively
- SPDIF features
 - 8, 16, 18, 20 and 24 bit audio sample data sizes supported
 - DMA transfer mode supported
 - Stop serial clock supported
 - Programmable Interrupt function supported
 - Support IEC60958 two-channel PCM audio
 - Support IEC61937 multi-channel compressed audio
 - Support consumer mode and only support transmitter mode
 - Profession mode is not supported
 - The User data bit is '0' as it is not supported in the chip
 - Support sampling frequency from 32kHz to 192kHz
- PCM interface
 - Data starts with the frame PCMSYN or one PCMCLK later
 - Support three modes of operation for PCM
 - Short frame sync mode
 - Long frame sync mode
 - Multi-slot mode
 - Data is transferred and received with the MSB first
 - Support master mode and slave mode
 - The PCM serial output data, PCMDOUT, is clocked out using the rising edge of the PCMSCLK
 - The PCM serial input data, PCMDIN, is clocked in on the falling edge of the PCMSCLK
 - 8/16 bit sample data sizes supported
 - DMA transfer mode supported
 - Two FIFOs for transmit and receive respectively with 16 samples capacity in every direction
- Internal CODEC Interface
 - 24 bits ADC and DAC
 - Headphone load up to 16 Ohm
 - Sample frequency supported: 8k, 11.025k, 12k, 16k, 22.05k, 24k, 32k, 44.1k, 48k, and 96k
 - Stereo line input
 - DAC to HP path: Power consumption: 17.6mW, THD: -65dB @17.6mW /16Ohm
 - DAC to stereo line output path @10kOhm: SNR: 100dB A-Weighted, THD: -80dB @FS-1dB
 - Line input to ADC path: SNR: 90dB A-Weighted, THD: -80dB @FS-1dB

- Separate power-down modes for ADC and DAC path with several shutdown modes
- Reduction of audible glitches systems: Pop Reduction system, Soft Mute mode
- Output short circuit protection
- Embedded low noise Linear Regulator
- 4 MIC in path or 4 line in path Maximum (Total 4 analog input)
- Support Digital MIC

1.2.7 Memory Interface

- DDR Controller
 - Support DDR2, DDR3, mobile DDR (LPDDR), LPDDR2 memory, up to 800Mbps
 - Support x16 and x32 external DDR data width
 - Asynchronize to system bus and each port.
 - Support clock-stop mode
 - Support auto-refresh and self-refresh
 - Support power-down mode and deep-power-down mode
 - Programmable DDR timing parameters
 - Programmable DDR row and column address width and order
- Static memory interface
 - Support 6 external chip selection CS6~1#. Each bank can be configured separately
 - The size and base address of static memory banks are programmable
 - Direct interface to 8-bit bus width external memory interface devices or external static memory to each bank. Read/Write strobe setup time and hold time periods can be programmed and inserted in an access cycle to enable connection to low-speed memory
 - Wait insertion by WAIT pin
 - Automatic wait cycle insertion to prevent data bus collisions in case of consecutive memory accesses to different banks, or a read access followed by a write access to the same bank
- NAND flash interface
 - Support on CS6~CS1, sharing with static memory bank6~bank1
 - Support both of conventional NAND flash memory and Toggle NAND flash memory
 - Support most types of NAND flashes, 8-bit data access, 512B/2K/4K/8KB/16KB page size. For 512B page size, 3 and 4 address cycles are supported. For 2K/4K/8KB/16KB page size, 4 and 5 address cycles are supported
 - Support read/erase/program NAND flash memory
 - Support boot from NAND flash
- BCH Controller
 - Support up to 64-bit ECC encoding and decoding for NAND

1.2.8 System

- Clock generation and power management
 - On-chip 12/24/48MHZ oscillator circuit

- External 32.768KHZ input
- One four-chip phase-locked loops (PLL) with programmable multiplier
- CCLK, HHCLK, H2CLK, PCLK, H0CLK, DDR_CLK, VPU_CLK frequency can be changed separately for software by setting registers
- SSI clock supports 50M clock
- MSC clock supports 100M clock
- Functional-unit clock gating
- Shut down power supply for P0, VPU, GPU, GPS, P1
- Timer and counter unit with PWM output and/or input edge counter
 - Provide eight separate channels, six of them have input signal transition edge counter
 - 16-bit A counter and 16-bit B counter with auto-reload function every channel
 - Support interrupt generation when the A counter underflows
 - Three clock sources: RTCLK (real time clock), EXCLK (external clock input), PCLK (APB Bus clock) selected with 1, 4, 16, 64, 256 and 1024 clock dividing selected
 - Every channel has PWM output
- OS timer
 - 64-bit counter and 32-bit compare register
 - Support interrupt generation when the counter matches the compare register
 - Two clock sources: RTCLK (real time clock), HCLK (system bus clock) selected with 1, 4, 16, 64, 256 and 1024 clock dividing selected
- Interrupt controller
 - Total 64 interrupt sources
 - Each interrupt source can be independently enabled
 - Priority mechanism to indicate highest priority interrupt
 - All the registers are accessed by CPU
 - Unmasked interrupts can wake up the chip in sleep mode
 - Another set of source, mask and pending registers to serve for PDMA
- Watchdog timer
 - Generates WDT reset
 - A 16-bit Data register and a 16-bit counter
 - Counter clock uses the input clock selected by software
 - PCLK, EXTAL and RTCCLK can be used as the clock for counter
 - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software
- Direct memory access controllers
 - Support up to 32 independent DMA channels
 - Descriptor or No-Descriptor Transfer mode compatible with previous JZ SOC
 - A simple Xburst-1 CPU supports smart transfer mode controlled by programmable firmware
 - Transfer data units: 1-byte, 2-byte, 4-byte, 16-byte, 32-byte, 64-byte, 128-byte
 - Transfer number of data unit: $1 \sim 2^{24} - 1$
 - Independent source and destination port width: 8-bit, 16-bit, 32-bit
 - Fixed three priorities of channel groups: 0~3, highest; 4~11: mid; 12~31: lowest
 - A dedicated bus interface - BIF interconnects with on-chip BCH

- A dedicated bus interface - NIF interconnects with on-chip NEMC or off-chip NEMC.
- An extra INTC IRQ can be bound to one programmable DMA channel
- Dedicated Security ROM and Security RAM supporting enhanced security requirements.
- SAR A/D Controller
 - 7 Channels
 - Resolution: 12-bit
 - Integral nonlinearity: ± 1 LSB
 - Differential nonlinearity: ± 0.5 LSB
 - Resolution/speed: up to 2Msps
 - Max Frequency: 200k
 - Low power dissipation: 1.5mW(worst)
 - Support 4-wire and 5-wire touch panel measurement (Through pin XP, XN, YP, YN and AUX2)
 - Support multi-touch detect
 - Support write control command by software
 - Support voltage measurement (Through pin VBAT)
 - Support two auxiliary input (Through pin AUX1, AUX2)
 - Single-end and Differential Conversion Mode
 - Auto X/Y, X/Y/Z1/Z2 and X/Y/Z1/Z2/X2/Y2 position measurement
 - Support external touch screen controller
 - Pin Description
- RTC (Real Time Clock)
 - Need external 32768Hz oscillator for 32k clock generation
 - RTCLK selectable from the oscillator or from the divided clock of EXCLK, so that 32k crystal can be absent if the hibernating mode is not needed
 - 32-bits second counter
 - Programmable and adjustable counter to generate accurate 1 Hz clock
 - Alarm interrupt, 1Hz interrupt
 - Stand alone power supply, work in hibernating mode
 - Power down controller
 - Alarm wakeup
 - External pin wakeup with up to 2s glitch filter

1.2.9 Peripheral

- Transport stream slave interface
 - Support both parallel mode and serial mode for TS data transfer
 - TSDI0 or TSDI7 can be used to transfer data in serial mode
 - The order of data in one byte supports LSB at first or MSB at first
 - The order of data in one word supports LSB at first or MSB at first
 - Input control signals and data can be either active high or active low
 - Support using either positive or negative edge of TSCLK

- Support PID filtering function
- Up to 33 PID filters can be used when PID filtering function is enabled
- Support adding data 0 before or after transport stream
- Support master DMA transmission
- General-Purpose I/O ports
 - Each port can be configured as an input, an output or an alternate function port
 - Each port can be configured as an interrupt source of low/high level or rising/falling edge triggering. Every interrupt source can be masked independently
 - Each port has an internal pull-up or pull-down resistor connected. The pull-up/down resistor can be disabled
 - GPIO output 6 interrupts, 1 for every group, to INTC
- SMB Controller
 - Two-wire SMB serial interface – consists of a serial data line (SDA) and a serial clock (SCL)
 - Two speeds
 - Standard mode (100 Kb/s)
 - Fast mode (400 Kb/s)
 - Device clock is identical with pclk
 - Programmable SCL generator
 - Master or slave SMB operation
 - 7-bit addressing/10-bit addressing
 - 16-level transmit and receive FIFOs
 - Interrupt operation
 - The number of devices that you can connect to the same SMB-bus is limited only by the maximum bus capacitance of 400pF
 - APB interface
 - 5 independent SMB channels (SMB0, SMB1, SMB2, SMB3, SMB4)
- Two Synchronous serial interfaces (SSI0, SSI1)
 - 3 protocols support: National's Microwire, TI's SSP, and Motorola's SPI
 - Full-duplex or transmit-only or receive-only operation
 - Programmable transfer order: MSB first or LSB first
 - 128 entries deep x 32 bits wide transmit and receive data FIFOs
 - Configurable normal transfer mode or Interval transfer mode
 - Programmable clock phase and polarity for Motorola's SSI format
 - Two slave select signal (SSI_CE_ / SSI_CE2_) supporting up to 2 slave devices
 - Back-to-back character transmission/reception mode
 - Loop back mode for testing
- Five UARTs (UART0, UART1, UART2, UART3, UART4)
 - Full-duplex operation
 - 5-, 6-, 7- or 8-bit characters with optional no parity or even or odd parity and with 1, 1½, or 2 stop bits
 - 64x8 bit transmit FIFO and 64x11bit receive FIFO
 - Independently controlled transmit, receive (data ready or timeout), line status interrupts

- Internal diagnostic capability Loopback control and break, parity, overrun and framing-error is provided
- Separate DMA requests for transmit and receive data services in FIFO mode
- Supports modem flow control by software or hardware
- Slow infrared asynchronous interface that conforms to IrDA specification
- Three MMC/SD/SDIO controllers (MSC0, MSC1, MSC2)
 - Fully compatible with the MMC System Specification version 4.2
 - Support SD Specification 3.0
 - Support SD I/O Specification 1.0 with 1 command channel and 4 data channels
 - Consumer Electronics Advanced Transport Architecture (CE-ATA – version 1.1)
 - Maximum data rate is 50MBps
 - Support MMC data width 1bit ,4bit and 8bit
 - Built-in programmable frequency divider for MMC/SD bus
 - Built-in Special Descriptor DMA
 - Maskable hardware interrupt for SDIO interrupt, internal status and FIFO status
 - 128 x 32 built-in data FIFO
 - Multi-SD function support including multiple I/O and combined I/O and memory
 - IRQ supported enable card to interrupt MMC/SD controller
 - Single or multi block access to the card including erase operation
 - Stream access to the MMC card
 - Supports SDIO read wait, interrupt detection during 1-bit or 4-bit access
 - Supports CE-ATA digital protocol commands
 - Support Command Completion Signal and interrupt to CPU
 - Command Completion Signal disable feature
 - The maximum block length is 4096bytes
- USB 2.0 host interface
 - Open Host Controller Interface (OHCI)-compatible and USB Revision 1.1-compatible
 - High speed, Full speed and low speed
 - Embedded USB 2.0 PHY
- USB 2.0 OTG interface
 - Complies with the USB 2.0 standard for high-speed (480 Mbps) functions and with the On-The-Go supplement to the USB 2.0 specification
 - Operates either as the function controller of a high- /full-speed USB peripheral or as the host/peripheral in point-to-point or multi-point communications with other USB functions
 - Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
 - UTMI+ Level 3 Transceiver Interface
 - Soft connect/disconnect
 - 16 Endpoints:
 - Dedicate FIFO
 - Supports control, interrupt, ISO and bulk transfer
- OTP Slave Interface
 - Total 8K bits. Lower 192bits are read only, other higher bits are read-able and write-able
 - Support Security boot.

1.2.10 Bootrom

16kB Boot ROM memory

1.3 Characteristic

Item	Characteristic
Process Technology	40nm CMOS low power
Power supply voltage	General purpose I/O: 1.6~3.6V DDR I/O for DDR2: 1.8V± 0.1V DDR I/O for DDR3: 1.5V± 0.075V DDR I/O for DDR3L: 1.35V± 0.1V DDR I/O for DDR3U: 1.25V± 0.06V DDR I/O for LPDDR: 1.8V± 0.15V DDR I/O for LPDDR2: 1.2V± 0.1V RTC I/O: 1.8V~3.6V EFUSE programming: 2.5V± 10% Analog power supply 1: 2.5V± 10% Analog power supply 2: 3.3V± 10% Core: 1.1 -0.1/+0.2 V
Package	BGA390 17mm x 17mm x 1.1mm, 0.8mm pitch
Operating frequency	1.2GHz

Section 2

CORE FUNCTIONS

2 CPU

2.1 Overview

The CPU of this SOC is an SMP dual-core implementation with a unified L2 cache. It has following obvious features contrasting with past implementations of XBurst-1 micro-architecture.

- SMP dual core
- 32KB L1-Icache plus 32KB L1-Dcache for each core
- DCSC (Dual Core Schedule & Control), a dedicated global IRQ controller and dispatcher
- SW transparent global directory GDIR to support cache coherent protocol: MSI
- 512 KB unified L2 cache
- Dedicated UCA (uncacheable accelerate) cache attribute

2.2 CP0 Register Description

Note that this document just only describes new added CP0 registers which are reserved in the past implementations as well as some old ones with its part of bit field being changed in accordance with changing of hardware resources in this implementation.

2.2.1 Cores Control (CP0 Register 12, Select 2)

CORE_CTRL																CP0 Register 12, select 2																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	Reserved														SLEEP1M	SLEEP0M	Reserved						RPC1	RPC0	Reserved						SW_RST1	SW_RST0			
Rst	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	?	?	?	?	0	0	0	0	?	?	?	?	?	?	1	1	1	0

Bits	Name	Description	R/W
31:18	Reserved	Writing has no effect, read as zero.	R
17	SLEEP1M	Mask CORE1's clockdown. 0 – not mask; 1 – mask	RW
16	SLEEP0M	Mask CORE0's clockdown. 0 – not mask; 1 – mask	RW
15:10	Reserved	Writing has no effect, read as zero.	R
9	RPC1	Reset exception handler entry for CORE1. 1 – Dedicated reset entry; 0 – default reset entry (0xBFC00000)	RW
8	RPC0	Reset exception handler entry for CORE0. 1 – Dedicated reset entry; 0 – default reset entry (0xBFC00000)	RW
7:2	Reserved	Writing has no effect, read as zero.	R
1	SW_RST1	Software reset CORE1. 1 – keeps reset; 0 – not SW reset status	RW
0	SW_RST0	Software reset CORE0. 1 – keeps reset; 0 – not SW reset status	RW

2.2.2 Cores Status (CP0 Register 12, Select 3)

Cores_Status																	CP0 Register 12, select 3																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved														SLEEP1	SLEEP0	Reserved					IRQ1_P	IRQ0_P	Reserved					MIRQ1_P	MIRQ0_P			
Rst	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	0	0	?	?	?	?	?	?	?	0	0

Bits	Name	Description	R/W
31:18	Reserved	Writing has no effect, read as zero.	R
17	SLEEP1	Sleep status of CORE1. 0 – not sleep; 1 – sleep	R
16	SLEEP0	Sleep status of CORE0. 0 – not sleep; 1 – sleep	R
15:10	Reserved	Writing has no effect, read as zero.	R
9	IRQ1_P	Pending peripheral IRQ to CORE1. Software can only read the bit field. 1 – pending peripheral IRQ; 0 – no pending peripheral IRQ *1	R
8	IRQ0_P	Pending peripheral IRQ to CORE0. Software can only read the bit field. 1 – pending peripheral IRQ; 0 – no pending peripheral IRQ *1	R
7:2	Reserved	Writing has no effect, read as zero.	R
1	MIRQ1_P	Pending mailbox IRQ to CORE1 due to software writing mailbox1 register. Software can clear it to value 0 but setting value 1 has no effect. 1 – pending mailbox IRQ to CORE1; 0 – no pending *2	RW0
0	MIRQ0_P	Pending mailbox IRQ to CORE0 due to software writing mailbox0 register. Software can clear it to value 0 and setting value 1 has no effect. 1 – pending mailbox IRQ to CORE0; 0 – no pending *2	RW0

NOTES:

*1 - All kinds of peripheral IRQs are integrated in a dedicated interrupt controller named INTC which is outside of CPU, and finally only one global peripheral IRQ signal is issued from INTC. Therefore, for SMP, it had better choose no more than one core in COREn (n = 0, 1, ...) to accept such IRQ by setting correct mask bits defined in following CORE_REIM register.

*2 - If the corresponding mask bit (defined in following CORE_REIM register) has been set, the MIRQ signal can not be accepted by corresponding core

2.2.3 Reset Entry & IRQ Mask (CP0 Register 12, Select 4)

CORE_REIM																	CP0 Register 12, select 4															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENTRY														Reserved					IRQ1_M	IRQ0_M	Reserved					MIRQ1_M	MIRQ0_M				
Rst	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	?	?	?	?	?	?	0	0

Bits	Name	Description	R/W
31:16	Entry	Reset exception entry for a CORE when its RPC is set value 1. For example, when RPC1==1, and Entry == 0x9030, then the reset exception entry for CORE1 is 0x90300000.	RW
15:10	Reserved	Writing has no effect, read as zero.	R
9	IRQ1_M	Mask of IRQ from INTC to CORE1. 0 – mask; 1 – not mask	RW
8	IRQ0_M	Mask of IRQ from INTC to CORE0. 0 – mask; 1 – not mask	RW
7:2	Reserved	Writing has no effect, read as zero.	R
1	MIRQ1_M	Mask of mailbox IRQ to CORE1. 0 – mask; 1 – not mask	RW
0	MIRQ0_M	Mask of mailbox IRQ to CORE0. 0 – mask; 1 – not mask	RW

2.2.4 SPINLOCK (CP0 Register 12, Select 5)

SPINLOCK																CP0 Register 12, select 5																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																LOCK															
Rst	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:4	Reserved	Writing has no effect, read as zero.	R
3:0	LOCK	Lock value. Software can only clear it to zero, writing non-zero value to it is ignored.	RW0

2.2.5 SPINATOMIC (CP0 Register 12, Select 6)

SPINATOMIC																CP0 Register 12, select 6																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																VAL															
Rst	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

Bits	Name	Description	R/W
31:4	Reserved	Writing has no effect, read as zero.	R
3:0	VAL	For write operation, the value updating the register can update SPINLOCK at the same time if the value in SPINLOCK equals zero. Therefore, an atomic writing-nonzero-to-spinlock operation can be performed.	RW

2.2.6 Processor Identification (CP0 Register 15, Select 0)

Processor Identification															CP0 Register 15, select 0																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	PRID																																
Rst	0	0	1	1	1	1	1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	1	1

Bits	Name	Description	R/W
31:0	PRID	Reset value is 0x3EE1024F	R

2.2.7 EBase (CP0 Register 15, Select 1)

EBase															CP0 Register 15, select 1																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSB31 Config_30	EBASE															Reserved							CORE_ID								
Rst		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?		?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31	MSB31	Always read as 1, write is ignored	R
30	Config_30	If Config7.bit7 has been setting 1, the field can be modified, otherwise, it is read as 0	RW
29:12	EBASE	Bit field 29:12 of exception handler base when STATUS _{bev} = 0	RW
11:3	Reserved	Read as zero, write is ignored	R
2:0	Core_ID	ID (0, 1, ...) , identify each core, value was preset	R

2.2.8 Config1 Register (CP0 Register 16, Select 1)

Config1															CP0 Register 16, select 1																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	M	MMU Size					IS		IL		IA		DS		DL		DA		Reserved		PC	WR	CA	EP	EP							
Rst	1	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1	1	1	0	0	0	1	0	1	1

Bits	Name	Description	R/W
31	M	Hardwired to 1 to present the existence of Config2.	R
30:25	MMU Size	0x1F: 32 entries JTLB.	R

24:22	IS	Number of instruction cache sets per way. 0x1: 128	R
21:19	IL	Instruction cache line size. 0x4: 32 bytes	R
18:16	IA	Instruction cache associativity. 0x7: 8 way	R
15:13	DS	Number of data cache sets per way. 0x1: 128	R
12:10	DL	Data cache line size. 0x4: 32 bytes	R
9:7	DA	Data cache associativity. 0x7: 8 way	R
6:5	Reserved	Read as zero, write is ignored	R
4	PC	Performance Counter register implemented. 0: not implemented	R
3	WR	Watch registers implemented 1: implemented	R
2	CA	Code compression implemented 0: not implemented	R
1	EP	Debug EJTAG present. 1: present	R
0	FP	FPU implemented 1: implemented	R

2.2.9 Config7 Register (CP0 Register 16, Select 7)

Config7																CP0 Register 16, select 7																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PM_PCH																PM_EVT				PM_MD		PME	CFG_E30	PART_K	Reserved			ALLOCN	BTBINV	BTB_EN_	
Rst	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	?	?	?	0	0	0

Bits	Name	Description	R/W
31:16	PM_PCH	high 16 bits of monitored PC when use PMON, refer to PMON spec for detail	RW
15:12	PM_EVT	PMON event encoding, refer to PMON spec for detail	RW
11:9	PM_MD	Monitor mode of PMON, refer to PMON spec for detail	RW
8	PME	PMON enable, refer to PMON spec for detail	RW
7	CFG_E30	Permit modification of EBASE[30]. 1 – permit	RW
6	PART_K	Permit partial of kernel mode resources to be used by applications in user mode, including TCSM, CACHE	RW

		instructions. 1 – permit; 0 - forbid	
5:3	Reserved	Read as zero, write is ignored	R
2	ALLOCN	PREF 30 available. 0 – available; 1 – treat PREF 30 as a normal data prefetch.	RW
1	BTBINV	Writing 1 to the bit field to invalidate all entries of BTB. It is write only and read as zero, writing 0 has no effect.	W1
0	BTB_EN	BTB enable. 0 – enable; 1 – disable	RW

2.2.10 Mailbox0 (CP0 Register 20, Select 0)

CORE_MBOX0																CP0 Register 20, select 0																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSG0																															
Rst	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	MSG0	Message to CORE0. The action of Writing CORE_MBOX0 rises a positive pulse IRQ which can be automatically latched into CORE_Status.MIRQ0_P	RW

2.2.11 Mailbox1 (CP0 Register 20, Select 1)

CORE_MBOX1																CP0 Register 20, select 1																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSG1																															
Rst	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	MSG1	Message to CORE1. The action of Writing CORE_MBOX1 rises a positive pulse IRQ which can be automatically latched into CORE_Status.MIRQ1_P	RW

2.3 Application Notes

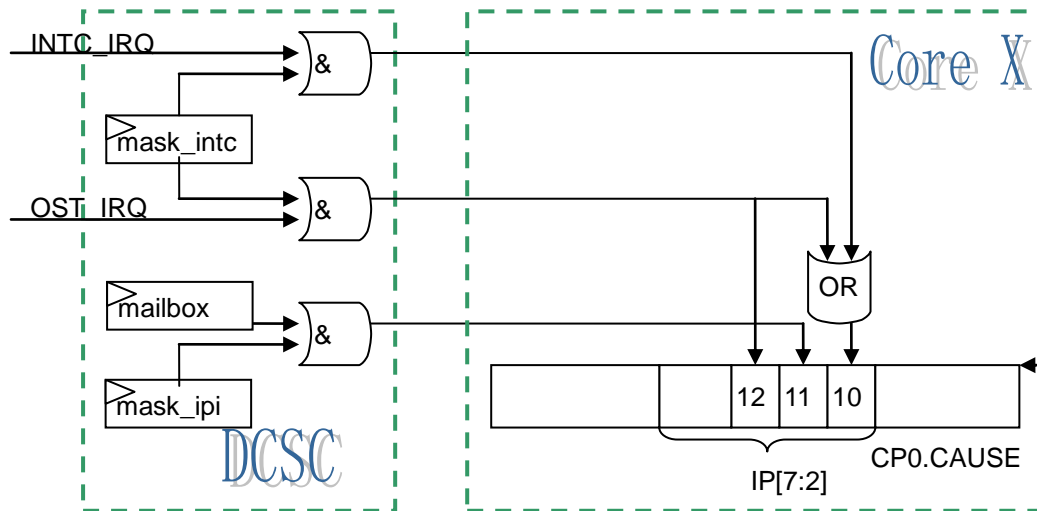
As a SMP dual-core implementation of XBurst-1, for system programmers, following points must be paid attention to:

1. A module named DCSC (Dual Core Schedule & Control) is a global IRQs (peripheral and IPI) controller and dispatcher, its relative spec CoreScheduler_SMP must be referenced first.
2. Both L1 I-cache and L1 D-cache are 32KB other than 16KB. Looking up CP0.config1 can get the correct information. Changing cache attribute 1 from write-through to uncacheable accelerate.
3. System OST IRQ now is separated from INTC, it is wired to each core's IP[4]
4. TLB now is completely compatible with MIPS 4KE, no default incompatible setting exists
5. MIPS32 Release2 instructions had been available since JZ4770.
6. Self-defined JTAG instructions are added for EJTAG debug of multiple cores.

2.3.1 Cache Attribute

Encoding	Description	Recommended usage
0	Write Through. Read miss with allocate Write miss without allocate	Not recommended
1	Uncacheable accelerate. Uncacheable for both read and write, but write is bufferable. That is, write buffer in CPU can gather uncacheable write data belonging to the same 32-byte block (address aligned at 32) , then sends them out with burst-write operation. This operation is running in the background.	Better for scenario of shared memory for which CPU is write-only and other device is read-only
2, 7	Uncacheable Both read and write are uncacheable	Memory mapped register's operation
3, 4, 5, 6	Write Back. Read miss with allocate Write miss with allocate	Most normal cases

2.3.2 Mapping of IRQs observed by CPU core



INTC_IRQ: original IRQ from external INTC, can be masked by corresponding mask bit in the DCSC, it is wired to each core's IP[2]

OST_IRQ: individual system OST IRQ, also can be masked by the mask bit for INTC_IRQ in the DCSC, it is wired to each core's IP[4] and IP[2]

IRQ of mailbox: IPI IRQ, can be masked by corresponding mask bit in the DCSC, it is wired to each core's IP[3].

Must treat the priority of IRQs visible for a core as the sequence: IP[4] > IP[3] > IP[2].

2.3.3 EJTAG debug for multiple cores

Four self-defined instructions are added. They are:

IR_EN_CORE0 ('b10100)

IR_EN_CORE1 ('b10101)

IR_EN_CORE2 ('b10110)

IR_EN_CORE3 ('b10111)

For the SMP implementation, only one core can be debugged through EJTAG at any time. CORE0 can be debugged through EJTAG by default. To toggle among cores, use above instructions before using EJTAGBOOT

3 VPU

Video Processing Unit (VPU) core in this chip is dedicated for video decoding and encoding. VPU embeds an XBurst[®] CPU core (named AUX in VPU) and application specified hardware accelerators for common video compress/decompress algorithms, which includes Stream Parser, Motion Compensation, Motion Estimation, Quant/Inverse Quant, DCT/Inverse DCT and De-block engines. Also VPU scheduler (named SCH in VPU) is embedded so that AUX can be standby and SCH would manage the whole VPU control and data flow for most popular video formats. That is to say, we can get preferable trade-off of high performance and low power consumption of most popular video formats by SCH and flexible programmability for other video formats or even some other arithmetic such as image processing by AUX.

Key standards performance of VPU in the chip:

- MPEG-2 decoding up to 1080P 60fps
- VC-1 decoding up to 1080P 60fps
- H.264 decoding up to 1080P 60fps
- VP8 decoding up to 1080P 60fps
- MPEG-4 decoding up to 1080P 30fps
- RV9 decoding up to 1080P 30fps
- H.264 encoding up to 1080P 30fps
- JPEG compressing/decompressing up to 100Mega-pixels per second (baseline)

3.1 Block Diagram

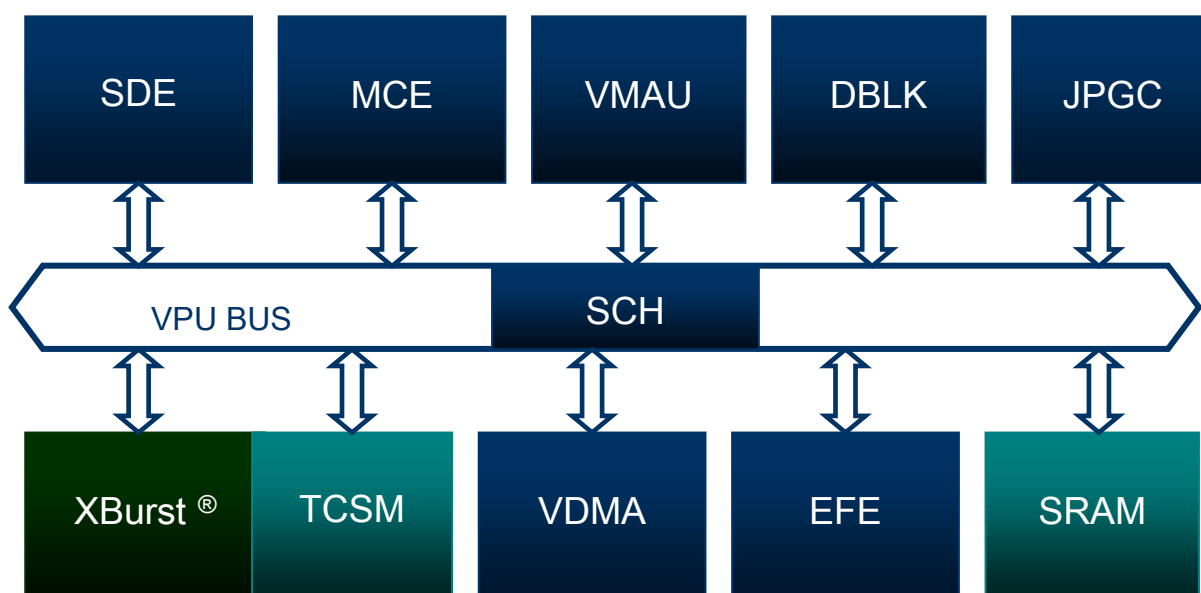


Figure 3-1 VPU Block Diagram

3.2 Features of VPU

Table 3-1 VPU Features

Item	Features
XBurst [®] core(AUX)	<ul style="list-style-type: none"> ● XBurst-1 CPU <ul style="list-style-type: none"> – Industry standard RISC instruction set – 32 32-bit general purpose registers, no shadow GPR – Physical address accessing directly ● Media Extension Unit (MXU) <ul style="list-style-type: none"> – Ingenic SIMD instruction set II – fully pipelined
Tightly Coupled Sharing Memory (TCSM)	<ul style="list-style-type: none"> ● TCSM <ul style="list-style-type: none"> – Size: 32K bytes – AHB slave interface supports external DMA access <p>NOTE: TCSM is coupled with VPU XBurst[®] core internally.</p>
Scratch RAM (SRAM)	<ul style="list-style-type: none"> ● Size: 16K bytes ● AHB slave interface supports external DMA access
VPU Gerenal Purpose DMA(VDMA)	<ul style="list-style-type: none"> ● Descriptor based DMA ● VPU self automatic configure supported
Parser (SDE)	<p>Parser is a bit-stream engine (named as SDE) in VPU</p> <ul style="list-style-type: none"> ● H.264 Context adaptive binary arithmetic (CABAC) decoding support, PCM not supported (treated as bitstream error) ● H.264 Context adaptive variable length (CAVLC) decoding support, PCM not supported (treated as bitstream error) ● VC-1 bitstream decoding support ● MPEG2 bitstream decoding support ● VP8 bitstream decoding support ● H.264 Context adaptive binary arithmetic encoding support
Motion (MCE)	<p>Motion serves as a COMBO engine of compensation and estimation (named as MCE) in VPU</p> <ul style="list-style-type: none"> ● Reference data cache embed ● Descriptor based task fetching ● Programmable processing size from 2x2 to 16x16 (in estimation the size is from 4x4 to 16x16) ● Programmable interplation filter from 2-tap to 8-tap ● Programmable sub-pixel accuracy from 1/2-pixel to 1/8-pixel (in estimation searching accuracy is supported from integer to 1/4-pixel) ● Interlaced mode support ● Intensity compensation support ● Weighted prediction support ● Automatic rotation support for rotated referenced pictures

	<ul style="list-style-type: none"> ● Automatic expanding support for outside frame's reference ● Configurable searching strategy in estimation
Recover (VMAU)	<p>Recover is a Matrix Arithmetic Unit in VPU (named as VMAU), it serves for pixel's recovery and reconstruction during video decoding and encoding.</p> <ul style="list-style-type: none"> ● Descriptor based task fetching ● Configurable format intra prediction support ● Configurable format Inverse quant support ● Configurable format IDCT support ● Residual add for pixel's recovery ● Estimation subtract and pixel reconstruction in encoding flow
Deblock (DBLK)	<ul style="list-style-type: none"> ● Descriptor based task fetching ● RealVideo(RV9) in loop filter support ● H.264 in loop filter support, FMO/MBAFF not supported ● VC-1 deblock support ● VP8 deblock support ● write buffer for DMA out
Encoder front-end (EFE)	<ul style="list-style-type: none"> ● YUV RAW fetching and delivering
JPEG codec (JPGC)	<ul style="list-style-type: none"> ● baseline profile JPEG compressing/decompressing ● quant table configurable ● huffman table configurable
Scheduler (SCH)	<p>Scheduler is a special unit which is used to manage internal functional engines' handshake.</p> <ul style="list-style-type: none"> ● 4 programable channels ● 2 programable groups
Translation look-aside buffer (TLB)	<ul style="list-style-type: none"> ● 4Kbyte based pagesize mapping ● TLB miss automatic entry filling
Interrupt control center (INTC)	<ul style="list-style-type: none"> ● Task end interrupt (frame/slice end) ● TLB error interrupt ● Bitstream error interrupt ● VDMA ACFG error interrupt

3.3 Internal physical address base definition

Table 3-2 VPU Internal physical address base definition

Module	Physical address base
SCH	0x1320_0000
AUX	0x132A_0000
TCSM	0x132C_0000
SRAM	0x132F_0000
VDMA	0x1321_0000

EFE	0x1324_0000
MCE	0x1325_0000
DBLK	0x1327_0000
VMAU	0x1328_0000
SDE	0x1329_0000
JPGC	0x132E_0000

3.4 AUX

3.4.1 Register Definition

3.4.1.1 Control and Status

AUX_CTRL																								0x0										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	SLEEP															Reserved										BTB_INV	Reserved				MIRQ_EN	NMI_DIS	SW_NMI	SW_RST
RST	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0	0	1	

Bits	Name	Description	R/W
31	SLEEP	AUX sleep status. 1: sleep; 0: no sleep.	R
30:9	Reserved	Writing has no effect, read as zero.	R
8	BTB_INV	Writing 1 can invalid BTB. Writing 0 has no effect, read as zero.	W
7:4	Reserved	Writing has no effect, read as zero.	R
3	MIRQ_EN	1: enable message IRQ. 0: disable.	RW
2	NMI_DIS	1: NMI only wakes up AUX from sleep status 0: NMI wakes up AUX and switch PC to 0xF4000000	RW
1	SW_NMI	Nonmaskable IRQ (NMI). Writing 1 to the field triggers a NMI pulse to AUX. Writing 0 has no effect, read as zero.	W
0	SW_RST	Software reset. 1: let AUX keep at reset status; 0: do not reset.	RW

NOTES:

- 1 When NMI or IRQ or RESET exception occurs, AUX resumes from PC 0xF4000000.
- 2 When AUX wakes up by an NMI meanwhile NMI_DIS is 1, AUX just resumes from the next PC of the WAIT instruction.

3.4.1.2 SPINLOCK

AUX_SPINLK																0x4																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														LOCK	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
30:2	Reserved	Writing has no effect, read as zero.	R
1:0	LOCK	Lock status.	RW

AUX_SPIN1																0x8																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														SPIN1	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
30:2	Reserved	Writing has no effect, read as zero.	R
1:0	SPIN1	Reading SPIN1 triggers following special hardware operations. First, value of AUX_SPINLK will be checked, if the value equals zero, the value of SPIN1 will overwrite AUX_SPINLK immediately, otherwise, AUX_SPINLK keeps unchanged. Then reading AUX_SPINLK instead of SPIN1 supplies the final read result. Writing SPIN1 is a normal write operation.	RW

AUX_SPIN2																0xC																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														SPIN2	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
30:2	Reserved	Writing has no effect, read as zero.	R
1:0	SPIN2	The operations for SPIN1 also fit SPIN2 except the role of SPIN1 should be replaced by SPIN2.	RW

AUX_MIRQP																0x10																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														MIRQP	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:1	Reserved	Writing has no effect, read as zero.	R
0	MIRQP	Pending status of MIRQ (message IRQ to CORE) which can only be set to 1 by HW and be reset to 0 by SW. This pending IRQ is routing to main CPU core.	RW

AUX_MSG																0x14																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MESG																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:0	MESG	If AUX_CTRL.MIRQ_EN is value 1, writing the register raises an IRQ routing to the main CPU core meanwhile the AUX_MIRQP is set to 1 by HW automatically. The IRQ then keeps active until the register AUX_MIRQP is cleared to 0 by SW.	RW

CORE_MIRQP																0x18																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														MIRQP	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:1	Reserved	Writing has no effect, read as zero.	R
0	MIRQP	Pending status of MIRQ (message IRQ to AUX) which can only be set 1 by HW and be clear to 0 by SW. This pending IRQ is routing to AUX.	RW

	CORE_MSG																																0x1C
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	MESG																																
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	

Bits	Name	Description	R/W
31:0	MESG	If AUX_CTRL.MIRQ_EN is value 1, writing the register raises an IRQ routing to the AUX. The IRQ then keeps active until the register CORE_MIRQP is cleared to 0 by SW.	RW

3.5 TCSM/SRAM

TCSM0/TCSM1/SRAM serves as the VPU control flow and data flow’s communication between XBurst® CPU core with specified algorithm hardware accelerators and different hardware accelerators as well.

3.5.1 TCSM/SRAM space usage

Table 3-3 TCSM/SRAM space usage

	XBurst® AUX	HW accelerator
TCSM	0xF400_0000 ~ 0xF400_7FFF	0x132C_0000 ~ 0x132C_7FFF
SRAM	0x132F_0000 ~ 0x132F_3FFF	0x132F_0000 ~ 0x132F_3FFF

NOTES:

1. TCSM serves as AUX’s instruction and data memory and VPU inner HW accelerator data sharing space as well.
2. SRAM should not be accessed when it is in using by VPU internally.

3.6 Video Acceleration Block

Please refer to relative programming manual documents.

4 GPU

4.1 3D graphics

4.1.1 Introduction

SGX is a new generation of programmable PowerVR graphics and video IP cores. The PowerVR SGX architecture is scalable and can target all market segments from mainstream mobile devices to high-end desktop graphics. PowerVR SGX540 is a core designed for feature phone / PDA / handheld gaming applications.

PowerVR SGX efficiently processes a number of differing multimedia data types concurrently:

- Pixel Data
- Vertex Data
- Video Data
- General Purpose Processing

4.1.2 PowerVR SGX Architecture - Key Features

- 3D graphics, vector graphics and video encode and decode supported on common hardware
- Tile based architecture
- Universal Scalable Shader Engine – multi-threaded engine incorporating Pixel and Vertex Shader functionality
- Advanced Shader Feature Set – in excess of Microsoft VS3.0, PS3.0 & OGL2.0
- Industry standard API support – Direct3D Mobile, OGL-ES 1.1 and 2.0, OpenVG 1.1, OpenMax
- Fine grained task switching, load balancing and power management
- Advanced geometry DMA driven operation for minimum CPU interaction
- Programmable high quality image anti-aliasing
- Fully virtualised memory addressing for OS operation in a unified memory architecture
- Standard master and slave AXI bus interfaces

4.1.3 SGX540 – 3D Features

- Deferred Pixel Shading
- On chip tile floating point depth buffer
- 8-bit Stencil with on chip tile stencil buffer
- 8 parallel depth/stencil tests per clock
- Scissor test
- Texture support
 - Cube Map
 - Projected Textures
 - Non square Textures

- Texture Formats
 - o RGBA 8888,565,1555,1565
 - o Mono chromatic 8, 16, 16f, 32f, 32int
 - o Dual channel, 8:8, 16:16, 16f:16f
 - o Compressed Textures PVR-TC1, PVR-TC2, ETC1
 - o Programmable support for all YUV formats
- Resolution Support
 - o Frame buffer max size = 2048 x 2048
 - o Texture max size = 2048 x 2048
- Texture Filtering
 - o Bilinear, Trilinear, Anisotropic
 - o Independent min and mag control
- Anti-aliasing
 - o 4x Multisampling
 - o Programmable sample positions
- Indexed Primitive List support
 - o Bus mastered
- Programmable vertex DMA
- Render to texture
 - o Including twiddled formats
 - o Auto MipMap generation
- Multiple on chip render targets (MRT) – dependent on availability of on chip SoC memory used as intermediate data stores not included in the SGX core

4.1.3.1 Universal Scalable Shader Engine - Key Features

The USSE is the engine at the centre of the PowerVR SGX architecture and supports a broad range of instructions.

- Single programming model
 - Multi-threaded with 16 simultaneous execution threads and up to 64 simultaneous data instances
 - Zero-cost swapping in, and out, of threads
 - Cached program execution model – max program size 4096 instructions
 - Dedicated pixel processing instructions
 - Dedicated vertex processing instructions
 - Dedicated video encode/decode instructions
 - 2048 32-bit registers
- SIMD execution unit supporting operations in:
 - 32 Bit IEEE Float
 - 2-way 16 bit fixed point
 - 4-way 8 bit integer
 - 32bit bit-wise (logical only)
 - Static and Dynamic flow control

- Subroutine calls
- Loops
- Conditional branches
- Zero-cost instruction predication
- Procedural Geometry
 - Allows generation of primitives
 - Effective geometry compression
 - High order surface support
- External data access
 - Permits reads from main memory via cache
 - Permits writes to main memory
 - Data fence facility
 - Dependent texture reads

4.1.4 SGX540 – Performance

4 USSE Multi Threaded Execution Unit implementation of PowerVR SGX architecture

- 3D Peak Performance
 - Fill Rate 2 Pixels per clock
 - Vertex Rate 1 Triangle 10 clocks (Transform Only)
- Tile Size 16 x 16 pixels
- Programmable 4x multi sampling anti-aliasing (MSAA)
 - Rotated grid
 - Texturing performance unaffected
- Optimised memory efficiency using multi-level cache architecture
 - L0 cache – textures only – 4 x 64bit per USSE execution unit
 - L1 cache – general data – 1Kbyte per USSE execution unit
 - L2 cache – general data – 1Kbyte
 - Burst memory interface – minimum size = 4 DWord contiguous address

4.2 EXtreme 2D

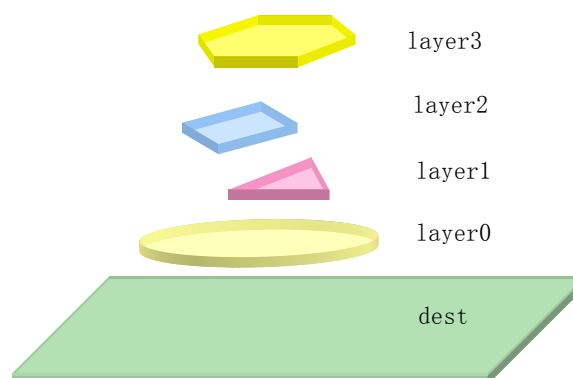
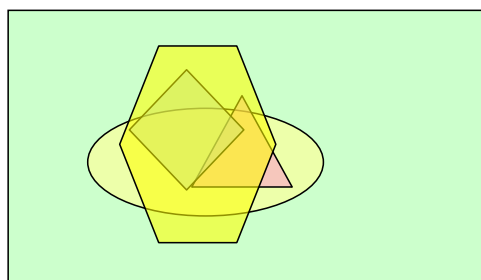
4.2.1 Overview

X2D is the unit which can take charge of rotation, scale, mirror, csc, and 4 layer osd.

4.2.1.1 Feature

- Location: AHB bus
- Input format
 - Separate frame: YUV /YCbCr (4:2:0)
 - Packaged data: RGB888, RGB565, RGB555, NV12, NV21, TileYUV
- Output data format

- ARGB888, XRGB888, RGB555, RGB565
- Color convention coefficient: configurable (CSC enable)
- Minimum input image size (pixel): 4x4
- Maximum input image size (pixel): 12288x12288 (12k x 12k)
- Maximum output image size (pixel)
 - Width : up to 12288
 - Height: up to 12288
- Image resizing
 - bi-cube zooming mode
- Image Clockwise 90, 180, 270 rotation
- Image horizontal and vertical mirror , same time with rotation
- 5 layers OSD



4.2.2 Registers Descriptions

The physical address base for the address-mapped registers of X2D is **0x13030000**.

The following table will show all the register address.

Table 4-1 register list

NAME	Offset	Descript
<u>GLB_CTRL</u>	0xF000	global controller
<u>GLB_STATUS</u>	0xF004	Global status
<u>GLB_TRIG</u>	0xF008	Global trigger
<u>TLB_BASE</u>	0xF00C	Base address of TLB (physical address map) table
<u>DHA</u>	0xF010	Descript chain's base address
<u>WDOG_CNT</u>	0xF018	Watch dog's counter
<u>LAY_GCTRL</u>	0xE000	Destination layers global control
<u>DST_BASE</u>	0xE004	Destination picture's base address
<u>DST_CTRL_STR</u>	0xE008	Destination picture's stride

<u>DST_GS</u>	0xE00C	Destination picture's size(height and width)
<u>DST_MSK_ARGB</u>	0xE010	Destination picture's ARGB mask
<u>DST_FMT</u>	0xE014	Destination picture format
Chanel 0		
<u>LAY0_CTRL</u>	0x0000	Source format control register
<u>LAY0_Y_ADDR</u>	0x0004	Source Y (or RGB) base address
<u>LAY0_U_ADDR</u>	0x0008	Source U (or UV) base address
<u>LAY0_V_ADDR</u>	0x000C	Source V base address
<u>LAY0_IN_FM_GS</u>	0x0010	Input Geometric Size (height and width)
<u>LAY0_STRIDE</u>	0x0014	Source frame stride (Y or RGB) and UV)
<u>LAY0_OUT_GS</u>	0x0018	Result frame size (height and width)
<u>LAY0_OOSFT</u>	0x001C	Left-up corner offset from Destination picture's left-up corner (y-offset and x-offset)
<u>LAY0_RSZ_COEF</u>	0x0020	Resize Coefficients(h-resize-coefficient and v-resize coefficient)
<u>LAY0_BK_ARGB</u>	0x0024	Background aRGB
Chanel 1		
<u>LAY1_CTRL</u>	0x1000	Source format control register
<u>LAY1_Y_ADDR</u>	0x1004	Source Y (or RGB) base address
<u>LAY1_U_ADDR</u>	0x1008	Source U (or UV) base address
<u>LAY1_V_ADDR</u>	0x100C	Source V base address
<u>LAY1_IN_FM_GS</u>	0x1010	Input Geometric Size (height and width)
<u>LAY1_STRIDE</u>	0x1014	Source frame stride (Y or RGB) and UV)
<u>LAY1_OUT_GS</u>	0x1018	Result frame size (height and width)
<u>LAY1_OOSFT</u>	0x101C	Left-up corner offset from Destination picture's left-up corner (y-offset and x-offset)
<u>LAY1_RSZ_COEF</u>	0x1020	Resize Coefficients(h-resize-coefficient and v-resize coefficient)
<u>LAY1_BK_ARGB</u>	0x1024	Background aRGB
Chanel 2		
<u>LAY2_CTRL</u>	0x2000	Source format control register
<u>LAY2_Y_ADDR</u>	0x2004	Source Y (or RGB) base address
<u>LAY2_U_ADDR</u>	0x2008	Source U (or UV) base address
<u>LAY2_V_ADDR</u>	0x200C	Source V base address
<u>LAY2_IN_FM_GS</u>	0x2010	Input Geometric Size (height and width)
<u>LAY2_STRIDE</u>	0x2014	Source frame stride (Y or RGB) and UV)
<u>LAY2_OUT_GS</u>	0x2018	Result frame size (height and width)
<u>LAY2_OOSFT</u>	0x201C	Left-up corner offset from Destination picture's left-up corner (y-offset and x-offset)
<u>LAY2_RSZ_COEF</u>	0x2020	Resize Coefficients(h-resize-coefficient and v-resize coefficient)
<u>LAY2_BK_ARGB</u>	0x2024	Background aRGB
Chanel 3		
<u>LAY3_CTRL</u>	0x3000	Source format control register
<u>LAY3_Y_ADDR</u>	0x3004	Source Y (or RGB) base address

LAY3_U_ADDR	0x3008	Source U (or UV) base address
LAY3_V_ADDR	0x300C	Source V base address
LAY3_IN_FM_GS	0x3010	Input Geometric Size (height and width)
LAY3_STRIDE	0x3014	Source frame stride (Y or RGB) and UV)
LAY3_OUT_GS	0x3018	Result frame size (height and width)
LAY3_OOSFT	0x301C	Left-up corner offset from Destination picture's left-up corner (y-offset and x-offset)
LAY3_RSZ_COEF	0x3020	Resize Coefficients(h-resize-coefficient and v-resize coefficient)
LAY3_BK_ARGB	0x3024	Background aRGB

4.2.2.1 Global control Register

GLB_CTRL																0xF000																										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
	reserved																SHARPL		BANK_SEL				L3_TLB				L2_TLB		L1_TLB		L0_TLB		CMD_TLB_EN		DMA_MOD		WDOG_EN		TLB_EN		IRQ_EN	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0										

Bits	Name	Description	R/W
31:14	Reserved	Writing has no effect, read as zero.	R
17:16	SHARPL	Bicube resize sharp level selector: 0: lowest sharp level 3: highest sharp level	RW
15:14	Reserved	Writing has no effect, read as zero.	R
13:12	BANK_SEL	ddr bank size selector: 1: 1K 2: 2K 3: 4K 0: reserved	RW
11:9	Reserved	Writing has no effect, read as zero.	R
8	L3_TLB	Lay3 source frame tlb enable. 0: disable, 1:enable	RW
7	L2_TLB	Lay2 source frame tlb enable. 0: disable, 1:enable	RW
6	L1_TLB	Lay1 source frame tlb enable. 0: disable, 1:enable	RW
5	L0_TLB	Lay0 source frame tlb enable. 0: disable, 1:enable	RW
4	CMD_TLB_EN	Command chain tlb endable: 0: disable, 1:enable	RW
3	DMA_MOD	DMA model selector. 0: no dma mode (not support) 1: dma mode (must)	RW
2	WDOG_EN	Watch dog enable. 1: enable; 0: disable	RW
1	DST_TLB_EN	Destination TLB enable. 1: enable; 0: disable	RW
0	IRQ_EN	Interrupt enable 1:enable; 0:disable	RW

4.2.2.2 Global Status Register

GLB_STATUS																0xF004																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	X2D_ID																reserved										WD_ERR	IRQ	GLB_STA			
RST	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	X2D_ID	Writing has no effect, read as 0xE003.	R
15:6	Reserved	Writing has no effect, read as zero.	R
5	WD_ERR	Watch dog error, it will happen when watch dog's time validate.	R
4	IRQ	Interrupt flag: 0: no interrupt generator 1: interrupt is happen.	R
3:0	GLB_STA	X2d global status. 0: free	R

4.2.2.3 Global trigger Register

GLB_TRIG																0xF008																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved																IRQ_CLR	RST	STOP	START												
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	Name	Description	R/W
3	IRQ_CLR	Clean the interrupt flag	W
2	RST	Reset X2D but register	W
1	STOP	Stop X2D immediately. 1: stop	W
0	START	Start X2D. 1: start	W

4.2.2.4 TLB address register

TLB_BASE																0xF00C																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	TLB_BASE																																0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	R/W
31:0	TLB_BASE	TLE base address	RW

4.2.2.5 Description chain address register

DHA																0xF010																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	DHA																																0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	R/W
31:0	DHA	Description chain's address.(Must DW align)	RW

*note: DHA will act only when the GLB_CTRL. DMA_MOD is 1.

4.2.2.6 Watch dog counter register

WDOG_CNT																0xF018																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDOG_CNT																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	WDOG_CNT	Watch dog counter	RW

4.2.2.7 Destination Layers global control

LAY_GCTRL																0xE000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DST_TILE															LMSK											Lay_num					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
16	DST_TILE	Destination result input and output Tile-mode selector. 1: Tile mode , 0: Plane mode	RW
2:0	Lay_num	Indicate the total layer numbers , it only action in DMA_MOD.	RW

4.2.2.8 Destination picture base address register

DST_BASE																0xE004																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DST_BASE																														0	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	DHA	Description chain's address.	RW

4.2.2.9 Destination picture control and stride register

DST_CTRL_STR																0xE008																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Alpha								Msk_en	preM	Giba_en	back_en	DST_STR																			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:24	Alpha	Destination global alpha value	RW

23	Msk_en	Destination will just act as an pure color background. 0: invalid 1: valid, this mode need not fetch data from external memory	RW
22	preM	Pre-multi indictor: 0: no pre-multi to RGB 1: pre-multi to RGB	RW
21	Glba_en	Destination buffer using global alpha, it will valid when back_en = 1. 0: pixel alpha 1: global alpha	RW
20	Back_en	Act the destination buffer as background picture	RW
19:0	DST_STR	Destination buffer's stride	RW

4.2.2.10 Destination picture Geometric size register

DST_GS																0xE00C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DST_HEIGHT																DST_WIDTH															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	DST_HEIGHT	Destination picture's height	RW
15:0	DST_WIDTH	Destination picture's width	RW

4.2.2.11 Destination picture mask value register

DST_MASK_ARGB																0xE010																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Alpha								R								G								B							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	ARGB	The value of the pixel when destination layer is masker	RW

4.2.2.12 Destination picture format register

DST_FMT																0xE014																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	APOS					RGB_FMT					RGB_ORD					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:5			
8	APOS	Alpha position 1: low position [7:0] , RGBA 0: high position [31:24] , ARGB	RW
7:6	Reserved	Writing has no effect, read as zero.	
5:4	RGB_FMT	0: ARGB888 1: XARGB888 2:RGB565 3:RGB555	RW
3	Reserved	Writing has no effect, read as zero.	
2:0	RGB_ORD	0: RGB 1:BGR 2:GRB 3:BRG 4:RBG 5:GBR	RW

4.2.2.13 Layer0 control Register

LAY0_CTRL																0x0000																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	OSD_MOD				APOS	RGBM								VMIR	HMIR	ROTA	GLB_ALPHA										Msk_en	CSCM	PREM	AL_MOD	IN_FMT		
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	R/W
31:28	OSD_MOD	3: clear 1: source 2: destination 0: source over 4: destination over 5: source in 6: destination in 7: source out 8: destination out 9: source atop A: destination atop	RW

		B: XOR	
27	APOS	Alpha position (only active when IN_FMT==0, alpha RGB888) 0: ARGB (A:[bit31~bit24]) 1: RGBA (A:[bit7~bit0])	
26:24	RGBM	RGB mode selector 0: RGB 1:BGR 2:GRB 3:BRG 4:RBG 5:GBR	R
23:20			
19	VMIR	Vertical mirror. 0:disable 1:enable	
18	HMIR	Horizontal mirror. 0:disable 1:enable	
17: 16	ROTA	Clockwise rotation selector 00: no rotation 01: 90 rotation 10: 180 rotation 11:270 rotation	R
15:8	GLB_ALPHA	Global alpha value (0 ~ 255) , valid when AL_MOD = 1 , or IN_FMT not ARGB888	RW
7			
6	MSK_EN	Lay will act as mask layer which just contain the same color. 0: Invalid 1: Valid, in this mod, the x2d will not need to fetch data from external memory, and all the pixel will be the background value	
5	CSCM	Color convention mode selector: 0: YUV => RGB 1: YCrCb => RGB	
4	PreM	The source pre-multi indicator: 0: No pre-multi to the RGB pixel 1: Pre-multi to the RGB pixel	
3	AL_MOD	Alpha mode selector. 1: global alpha 0:pixel alpha(Only when IN_FMT is ARGB888)	RW
2:0	IN_FMT	Indicates the source data format. 00: Alpha RGB888 01: RGB555 10: RGB565 11: YUV420 separate 100: Tile YUV420 101: NV12 110: NV21	RW

4.2.2.14 Layer0 input Y plane Address Register

LAY0_YADDR																0x0004																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAY0_YADDR																0	0														
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	LAY0_YADDR	In separated Frame case, it indicates the source Y (or R) data buffer's start address. In source package case, it indicates the start Address of the packaged Frame.	RW

4.2.2.15 Layer0 input U plane Address Register

LAY0_UADDR																0x0008																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAY0_UADDR																0	0														
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	LAY0_UADDR	The source U data buffer's start address of separated frame case.	RW

4.2.2.16 Layer0 input V plane Address Register

LAY0_VADDR																0x000C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAY0_VADDR																0	0														
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	LAY0_VADDR	The source V data buffer's start address of separated Frame case.	RW

4.2.2.17 Layer0 source frame Geometric Size Register

LAY0_SGS																0x0010																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IN_FM_H																IN_FM_W															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	IN_FM_H	The width of the input frame (Unit : pixel).	RW
15:0	IN_FM_W	The height of the input frame (Unit : pixel).	RW

4.2.2.18 Layer0 plane Stride Register

LAY0_STR																0x0014																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	UV_STR																Y_S																0	0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Bits	Name	Description	R/W
31:16	UV_STR	The line stride of the source UV data in the external(DDR) memory (Unit: byte) when YUV420. In YUV Tile, UV_STR should be the 1/16 to the stride of UV frame (UV_STR = UV_STR/16).	R
15:0	Y_S	The line stride of the source Y(or RGB) data in the external(DDR) memory (Unit: byte). In YUV Tile, Y_S should be the 1/16 to the stride of Y frame (Y_S = Y_S/16).	RW

4.2.2.19 Layer0 Output Geometric Size Register

LAY0_OGS																0x0018																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OUT_FM_H																OUT_FM_W															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	OUT_FM_H	The width of the output destination frame (Unit: pixel).	RW
15:0	OUT_FM_W	The height of the output destination frame (Unit: pixel).	RW

4.2.2.20 Layer0 Output start pos offset Register

LAY0_OOSF																0x001C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OUT_H_OFST																OUT_W_OFST															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	OUT_H_OFST	The horizontal offset in the output destination frame (Unit: pixel).	RW
15:0	OUT_W_OFST	The vertical offset in the output destination frame (Unit: pixel).	RW

4.2.2.21 Layer0 scale ratio Register

LAY0_RSZ																0x0020																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VRSZ																HRSZ															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	VRSZ	The scale ratio of vertical	RW
15:0	HRSZ	The scale ratio of horizontal	RW

*XRSZ = (uint16_t)((m/n)*512), m raw size, n: result size

4.2.2.22 Layer0 background ARGB Register

LAY0_BK_ARGB																0x0024																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Alpha								YR								UG								VB							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:24	Alpha	Background alpha	RW
23:16	YR	Background YR value (Y or R)	RW
15:8	UG	Background UG value (U or G)	RW
7:0	VB	Background VB value (V or B)	RW

4.3 Software stack

Structure X2D_CTRL_CHN

```

{
  Uint32_t  overlay_num;    // Total layer numbers and Destination-Tile-enable , see spec.23.2.7
  Uint32_t  * dst_addr;    // Destination picture address
  Uint32_t  dst_ctrl_str;  // Destination control information and destination stride , see spec.23.2.9
  Uint32_t  dst_gs;       // Destination picture's height and width
  Uint32_t  dst_bargb;    // Destination global argb
  Uint32_t  dst_fmt;      // Destination data format and rgb_mode
  X2D_OVERLAY_INFO layer_chain[4]; // Chain information of each 4 layer
}

```

Structure X2D_OVERLAY_INFO

```

{
  Uint32_t  lay_ctrl;      // Layer control information
  Uint32_t  Y_addr;       // Y or RGB source's address
  Uint32_t  U_addr;       // U source's address when YUV420
  Uint32_t  V_addr;       // V source's address when YUV420
  Uint16_t  lay_swidth;   // Y or RGB source's width (Unit : pixel)
  Uint16_t  lay_sheight;  // Y or RGB source's height (Unit : pixel)
  Uint16_t  lay_ystrid;   // Y or RGB source's stride
  Uint16_t  lay_uvstrid;  // UV source's stride
                          // (Unit : byte , when in Tile-YUV ,see spec.23.2.17)
  Uint16_t  lay_owidth;   // Destination's width
  Uint16_t  lay_oheight;  // Destination's height
  Uint16_t  lay_ox_offset; // Destination's x-offset from background's left-up corner.
  Uint16_t  lay_oy_offset; // Destination's y-offset from background's left-up corner.
  Uint16_t  lay_horiz_rsz_coef; // Current layer's horizontal resize coefficient
                          // RSZ = ( uint16_t )(m/n)*512), m raw size, n: result size
  Uint16_t  lay_verit_rsz_coef; // Current layer's vertical resize coefficient
  Uint32_t  lay_bk_argb;   // Current layer's background color or common alpha when layer
  mask enable
}

```

Section 3

DISPLAY/CAMERA/AUDIO

5 LCD Controller

5.1 Overview

The JZ integrated LCD controller has the capabilities to driving the latest TFT LCD panels. It also supports some special TFT panels used in consuming electronic products. The controller performs the basic memory based frame buffer to LCD panel data transfer through use of a dedicated DMA controller. And OSD is also supported for LCD controller.

Features:

- Basic Features
 - Support dual panel(HDMI & TFT, or HDMI & LVDS, or TFT & LVDS, or HDMI & SLCD, or SLCD & LVDS)
 - Display size up to 4kx2k@30Hz(BPP24) for single panel, 1920x1080@60Hz(BPP24) for dual panel
- Colors Supports
 - Encoded pixel data of 16, 18 or 24 BPP in TFT mode
 - Support up to 16,777,216 (16M) colors in TFT mode
 - Support 24/16 BPP compressed data
 - Support 24 BPP packed data
- Panel Supports
 - Support 16-bit parallel TFT panel(LCDC0 or LCDC1)
 - Support 18-bit parallel TFT panel(LCDC0 or LCDC1)
 - Support 24-bit serial TFT panel with 8 data output pins(LCDC0 or LCDC1)
 - Support 24-bit parallel TFT panel (LCDC0 or LCDC1)
 - Support Delta RGB panel(LCDC0 or LCDC1)
 - Support SLCD panel(LCDC0 or LCDC1)
 - Support LVDS Interface(just LCDC1)
 - Support HDMI 1.4a Interface(just LCDC0)
- OSD Supports
 - Supports two single color background
 - Supports two foregrounds, and every size can be set for each foreground
 - Supports one transparency for the whole graphic
 - Supports one transparency for each pixel in one graphic
 - Supports color key and mask color key
 - Supports porter-duff blending
- Image Enhancement

- Color space conversion: RGB to YCbCr, YCbCr to RGB
- Support Contrast, Brightness, Hue, Saturation control
- Support Visibility Enhance
- Support Dither
- Support Gamma Correction

5.2 Pin Description

Table 5-1 LCD Controller Pins Description

Name	I/O	Description
Lcd_pclk	Input/Output	Display device pixel clock
Lcd_vsync	Input/Output	Display device vertical synchronize pulse
Lcd_hsync	Input/Output	Display device horizontal synchronize pulse
Lcd_de	Output	Display device data enable Pin
Lcd_d[17:0]	Output	Display device data pins
lcd_lo6_o[5:0]	Output	Display device data pins use in 24 bit parallel mode.
Lcd_spl* ¹	Output	Programmable special pin for generating control signals
Lcd_cls* ¹	Output	Programmable special pin for generating control signals
Lcd_ps* ¹	Output	Programmable special pin for generating control signals
Lcd_rev* ¹	Output	Programmable special pin for generating control signals

NOTE: The mode and timing of special pin Lcd_spl, Lcd_cls, Lcd_ps and Lcd_rev can be seen in **part 5.7 LCD Controller Pin Mapping.**

5.3 Block Diagram

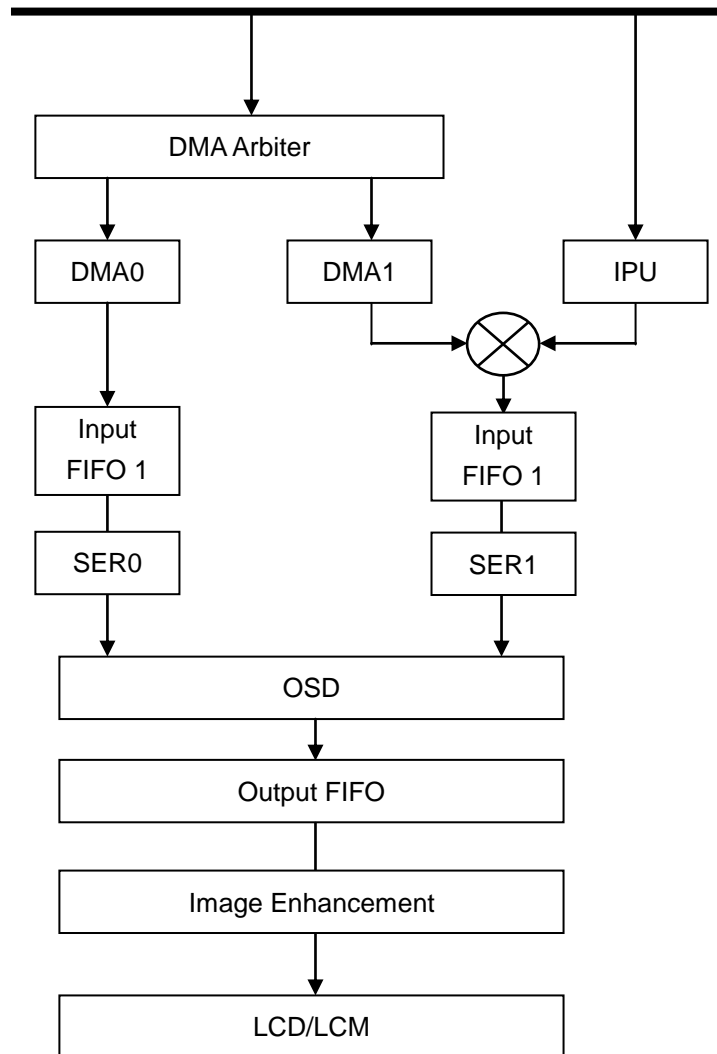


Figure 5-1 Block Diagram when use OSD mode

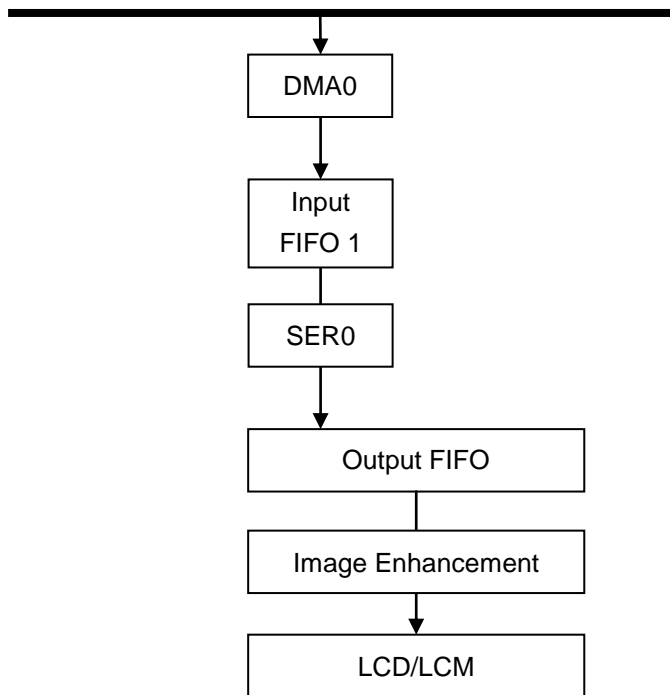


Figure 5-2 Block Diagram of TFT mode (not use OSD)

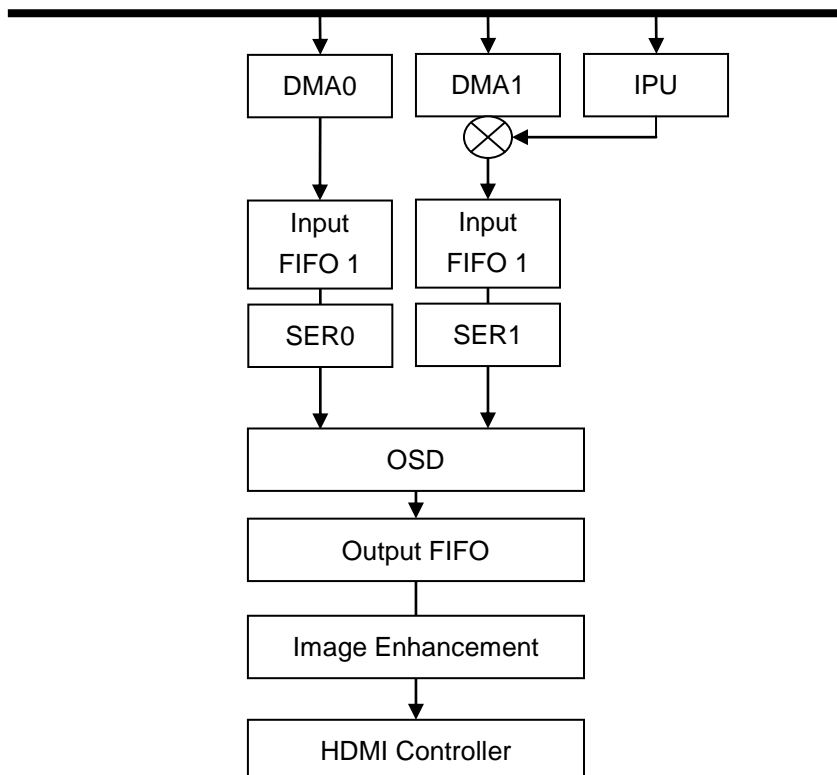


Figure 5-3 Block Diagram of HDMI interface

5.4 LCD Display Timing

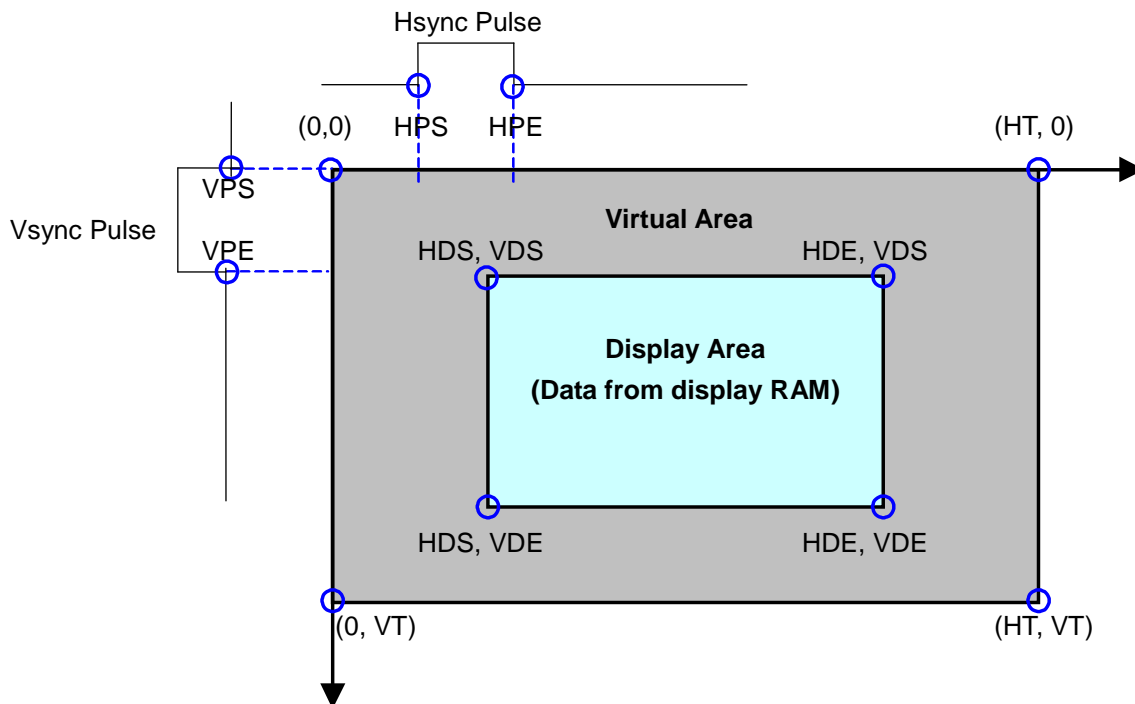


Figure 5-4 Display Parameters

NOTES:

- 1 VPS === 0
VSYNC pulse always start at point (0,0)
- 2 H: Horizontal V: Vertical T: Total
D: Display Area P: Pulse
S: Start point E: End point

In the (H, V) Coordinates:

- 1 The gray rectangle (0, 0) to (HT, VT) is "Virtual Area".
- 2 The blue rectangle (HDS, VDS) to (HDE, VDE) is "Display Area".
- 3 VPS, VPE defines the VSYNC signal timing. (VPS always be zero)
- 4 HPS, HPE defines the HSYNC signal timing.

All timing parameters start with "H" is measured in lcd_pclk ticks.

All timing parameters start with "V" is measured in lcd_hsync ticks.

This diagram describes the general LCD panel parameters, these can be set via the registers that describes in next section.

5.5 OSD Graphic

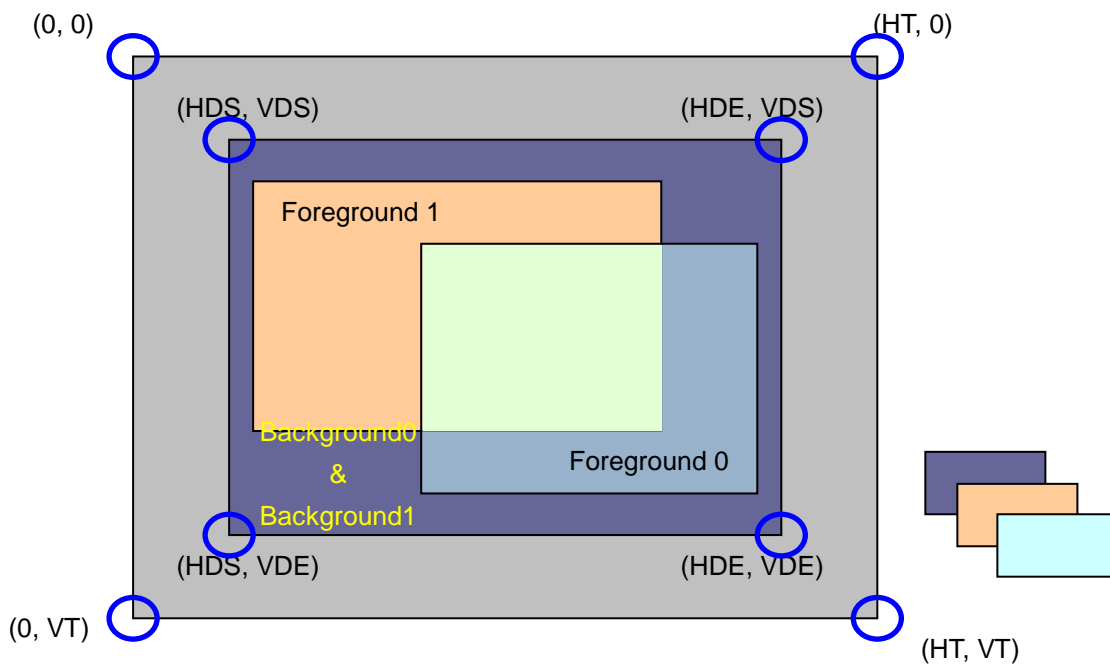


Figure 5-5 OSD Graphic

NOTES:

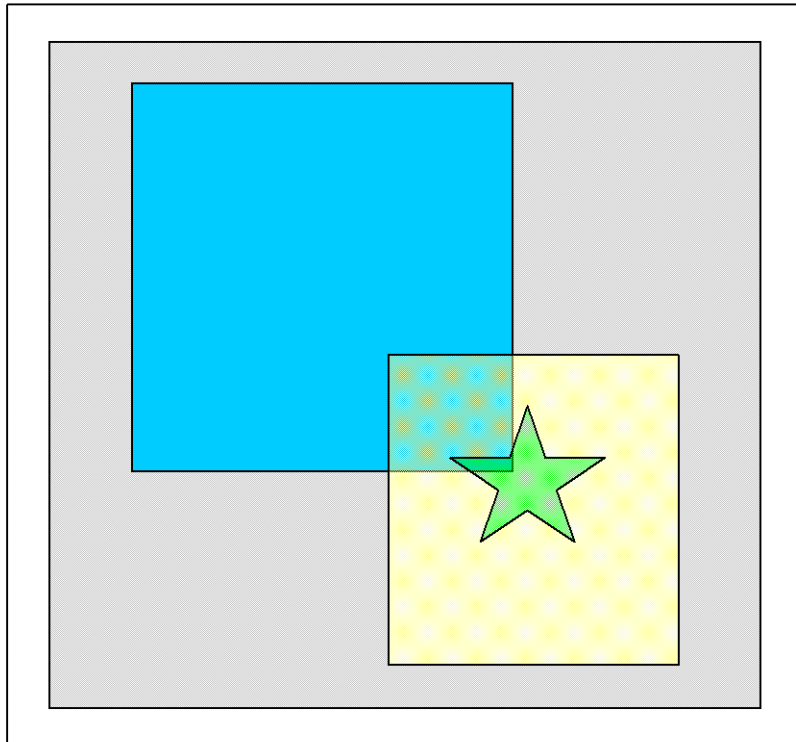
- 1 Background is one single color and the size is the full screen.
- 2 The size of foregrounds can be every size smaller than background.
- 3 The order of the graphic is as follows:
 - a Top layer: Foreground 0.
 - b Middle layer: Foreground 1.
 - c Bottom layer: Background0 & Background1.

5.5.1 Color Key

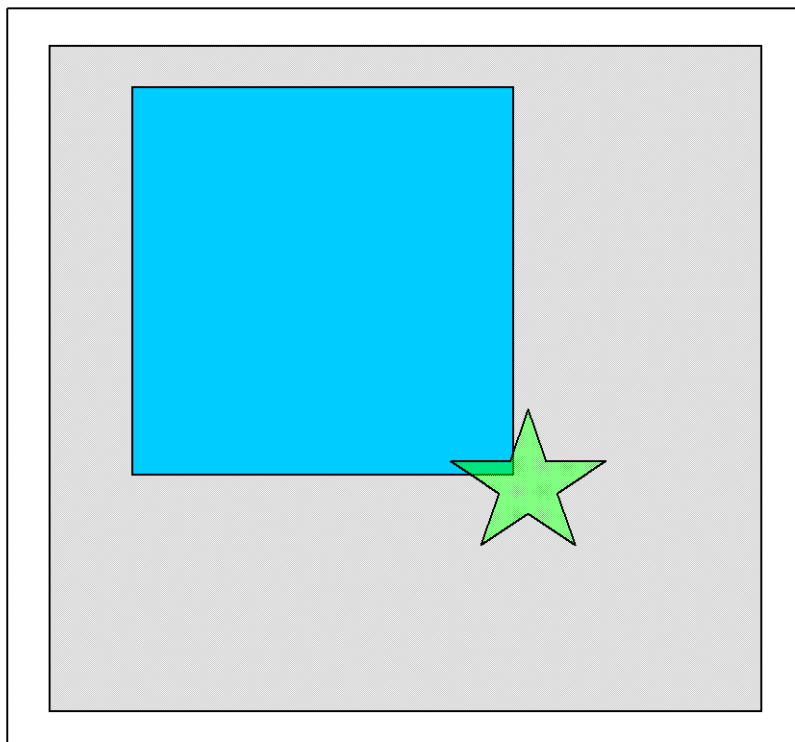
This function gives user a method to implement irregular display window. User can make foreground 0 and foreground 1 to different shape. The color key has two implements mode that called color key and mask color key.

Color Key mode is meant to mask a chosen color and show others.

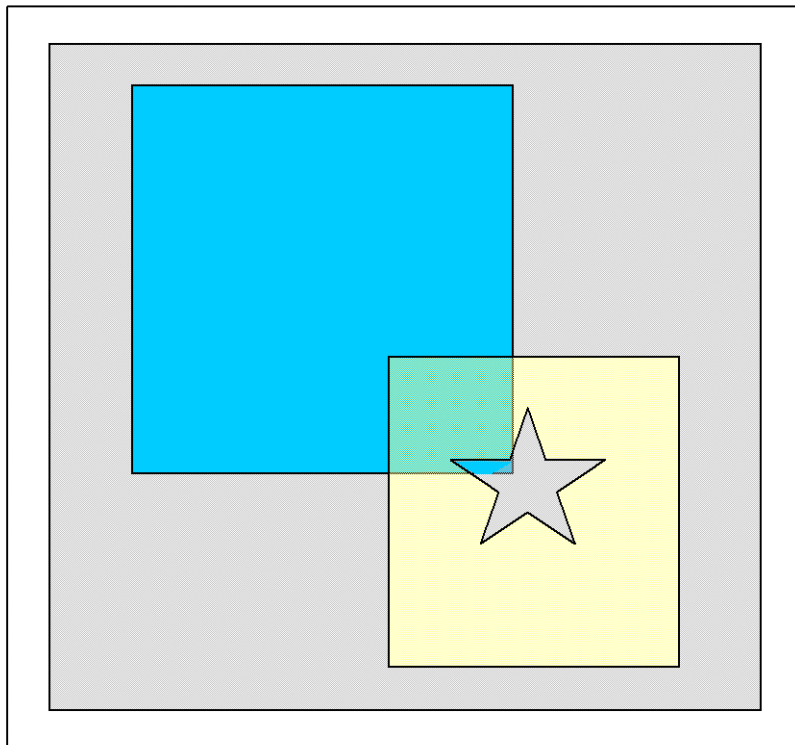
Mask Color Key mode is meant to only show a chosen color and mask others.



Not use color key function



Color key mode



Mask color key mode

5.6 Register Description

0x13050000 is the base address of lcdc0, 0x130a0000 is the base address of lcdc1.

Table 5-2 LCD Controller Registers Description

Name	RW	Reset Value	Address offset	Access Size
LCDCFG	RW	0x00000000	0x0000	32
LCDCTRL	RW	0x00000000	0x0030	32
LCDSTATE	RW	0x00000000	0x0034	32
LCDOSDC	RW	0x0000	0x0100	16
LCDOSDCTRL	RW	0x0000	0x0104	16
LCDOSDS	RW	0x0000	0x0108	16
LCDBG0	RW	0x00000000	0x010C	32
LCDBG1	RW	0x00000000	0x02C4	32
LCDKEY0	RW	0x00000000	0x0110	32
LCDKEY1	RW	0x00000000	0x0114	32
LCDALPHA	RW	0x00	0x0118	8
LCDIPUR	RW	0x00000000	0x011C	32
LCDRGBC	RW	0x0000	0x0090	16
LCDVAT	RW	0x00000000	0x000C	32
LCDDAH	RW	0x00000000	0x0010	32

LCDDAV	RW	0x00000000	0x0014	32
LCDXYP0	RW	0x00000000	0x0120	32
LCDXYP1	RW	0x00000000	0x0124	32
LCDSIZE0	RW	0x00000000	0x0128	32
LCDSIZE1	RW	0x00000000	0x012C	32
LCDVSYNC	RW	0x00000000	0x0004	32
LCDHSYNC	RW	0x00000000	0x0008	32
LCDPS ^{*1}	RW	0x00000000	0x0018	32
LCDCLS ^{*1}	RW	0x00000000	0x001C	32
LCDSPL ^{*1}	RW	0x00000000	0x0020	32
LCDREV ^{*1}	RW	0x00000000	0x0024	32
LCDIID	R	0x00000000	0x0038	32
LCDDA0	RW	0x00000000	0x0040	32
LCDSA0	R	0x00000000	0x0044	32
LCDFID0	R	0x00000000	0x0048	32
LCDCMD0	R	0x00000000	0x004C	32
LCDOFFS0	R	0x00000000	0x0060	32
LCDPW0	R	0x00000000	0x0064	32
LCDCNUM0/LCDPOS0	R	0x00000000	0x0068	32
LCDESSIZE0	R	0x00000000	0x006C	32
LCDDA1 ^{*2}	RW	0x00000000	0x0050	32
LCDSA1 ^{*2}	R	0x00000000	0x0054	32
LCDFID1 ^{*2}	R	0x00000000	0x0058	32
LCDCMD1 ^{*2}	R	0x00000000	0x005C	32
LCDOFFS1 ^{*2}	R	0x00000000	0x0070	32
LCDPW1 ^{*2}	R	0x00000000	0x0074	32
LCDCNUM1/LCDPOS1 ^{*2}	R	0x00000000	0x0078	32
LCDESSIZE1 ^{*2}	R	0x00000000	0x007C	32
LCDPCFG	RW	0x00000000	0x02C0	32
LCDDUALCTRL	RW	0x00000000	0x02C8	32
LCDENH_CFG	RW	0x00000000	0x0400	32
LCDENH_CSCCFG	RW	0x00000000	0x0404	32
LCDENH_LUMACFG	RW	0x00000000	0x0408	32
LCDENH_CHROCFG0	RW	0x00000000	0x040C	32
LCDENH_CHROCFG1	RW	0x00000000	0x0410	32
LCDENH_DITHERCFG	RW	0x00000000	0x0414	32
LCDENH_STATUS	RW	0x00000000	0x0418	32
LCDENH_GAMMA	RW	0x00000000	0x0800~ 0x0FFF	32
LCDENH_VEE	RW	0x00000000	0x1000~ 0x17FF	32

NOTES:

- 1 ^{*1}: These registers are only used for SPECIAL TFT panels.
- 2 ^{*2}: These registers are only used for DMA channel 1 in OSD mode for TFT panels.

5.6.1 Configure Register (LCDCFG)

LCDCFG																0x13050000, 0x130a0000																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	LCDPIN	TVEPEH		NEWDES		TVEN	RECOVER		PSM	CLSM	SPLM	REVM	HSYNM	PCLKM	INVDAT	SYNDIR	PSP	CLSP	SPLP	REVP	HSP	PCP	DEP	VSP	18/16	24							MODE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW						
31	LCDPIN ^{*1}	LCD PIN Select bit. It is used to choose the function of LCD PINS or SLCD PINS. The function of pins is as follows. <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <th style="width: 50%;">LCDPIN</th> <th style="width: 50%;">PIN SELECT</th> </tr> <tr> <td>0</td> <td>LCD PIN</td> </tr> <tr> <td>1</td> <td>SLCD PIN</td> </tr> </table>	LCDPIN	PIN SELECT	0	LCD PIN	1	SLCD PIN	RW
LCDPIN	PIN SELECT								
0	LCD PIN								
1	SLCD PIN								
30	TVEPEH	TVE PAL enable extra_halfline signal.	RW						
29		KEEP THIS BIT TO 0.	RW						
28	NEWDES	indicate use new 8 words descriptor or not. Set this bit to 1	RW						
27		KEEP THIS BIT TO 0.	RW						
26	TVEN	Indicate the terminal is LCD panel or TV.	RW						
25	RECOVER	Auto recover when output FIFO under run. 0: disable; 1: enable.	RW						
23	PSM	PS signal mode bit. 0: enabled; 1: disabled.	RW						
22	CLSM	CLS signal mode bit. 0: enabled; 1: disabled.	RW						
21	SPLM	SPL signal mode bit. 0: enabled; 1: disabled.	RW						
20	REVM	REV signal mode bit. 0: enabled; 1: disabled.	RW						
19	HSYNM	H-Sync signal polarity choice function. 0: enabled; 1: disabled.	RW						
18	PCLKM	Dot clock signal polarity choice function. 0: enabled; 1: disabled.	RW						
17	INVDAT	Inverse output data. 0: normal; 1: inverse.	RW						
16	SYNDIR	V-Sync and H-Sync direction. 0: output; 1: input.	RW						
15	PSP	PS pin reset state.	RW						
14	CLSP	CLS pin reset state.	RW						
13	SPLP	SPL pin reset state.	RW						
12	REVP	REV pin reset state.	RW						
11	HSP	H-Sync polarity. 0: active high; 1: active low.	RW						
10	PCP	Pix-clock polarity.	RW						

		0: data translations at rising edge 1: data translations at falling edge																																			
9	DEP	Data Enable polarity. 0: active high; 1: active low.	RW																																		
8	VSP	V-Sync polarity. 0: leading edge is rising edge 1: leading edge is falling edge	RW																																		
7	18/16	18-bit TFT Panel or 16-bit TFT Panel. This bit will be available when MODE [3:2] is equal to 0 and 24[6] is equal to 0. 0: 16-bit TFT Panel 1: 18-bit TFT Panel	RW																																		
6	24	Set this bit to 1 for 24-bit TFT Panel.	RW																																		
3:0	MODE	Display Device Mode Select/Output mode. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">LCD Panel</th> </tr> </thead> <tbody> <tr><td>0000</td><td>Generic 16-bit/18-bit Parallel TFT Panel</td></tr> <tr><td>0001</td><td>Special TFT Panel Mode1</td></tr> <tr><td>0010</td><td>Special TFT Panel Mode2</td></tr> <tr><td>0011</td><td>Special TFT Panel Mode3</td></tr> <tr><td>0100</td><td>Non-Interlaced TV out</td></tr> <tr><td>0101</td><td>Reserved</td></tr> <tr><td>0110</td><td>Interlaced TV out</td></tr> <tr><td>0111</td><td>Reserved</td></tr> <tr><td>1000</td><td>Reserved</td></tr> <tr><td>1001</td><td>Reserved</td></tr> <tr><td>1010</td><td>Reserved</td></tr> <tr><td>1011</td><td>Reserved</td></tr> <tr><td>1100</td><td>8-bit Serial TFT</td></tr> <tr><td>1101</td><td>LCM</td></tr> <tr><td>1110</td><td>Reserved</td></tr> <tr><td>1111</td><td>Reserved</td></tr> </tbody> </table>	LCD Panel		0000	Generic 16-bit/18-bit Parallel TFT Panel	0001	Special TFT Panel Mode1	0010	Special TFT Panel Mode2	0011	Special TFT Panel Mode3	0100	Non-Interlaced TV out	0101	Reserved	0110	Interlaced TV out	0111	Reserved	1000	Reserved	1001	Reserved	1010	Reserved	1011	Reserved	1100	8-bit Serial TFT	1101	LCM	1110	Reserved	1111	Reserved	RW
LCD Panel																																					
0000	Generic 16-bit/18-bit Parallel TFT Panel																																				
0001	Special TFT Panel Mode1																																				
0010	Special TFT Panel Mode2																																				
0011	Special TFT Panel Mode3																																				
0100	Non-Interlaced TV out																																				
0101	Reserved																																				
0110	Interlaced TV out																																				
0111	Reserved																																				
1000	Reserved																																				
1001	Reserved																																				
1010	Reserved																																				
1011	Reserved																																				
1100	8-bit Serial TFT																																				
1101	LCM																																				
1110	Reserved																																				
1111	Reserved																																				

NOTES:

*1:

LCDPIN	PIN25	PIN24	PIN23	PIN22	PIN21	PIN20	PIN19	PIN18	PIN17-0
0	LCD PCLK	LCD VSYNC	LCD HSYNC	LCD DE	LCD REV	LCD PS	LCD CLS	LCD SPL	LCD D [17:0]
1	SLCD CLK	SLCD CS	SLCD RS	--	--	--	--	--	SLCD D [17:0]

- 1 The direction of PIN25 is set by register LPCDR.LCS in CPM SPEC.
- 2 The direction of PIN23 and PIN23 are set by register LCDCFG.SYNDIR.

5.6.2 Control Register (LCDCTRL)

LCDCTRL																0x13050030, 0x130a0030																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	PINMD	BST				OUTRGB	OFUP											DACTE	EOFM	SOFM	OFUM	IFUM0	IFUM1	LDDM	QDM	BEDN	PEDN	DIS	ENA	BPP0			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW												
31	PINMD	This register set Pin distribution in 16-bit parallel mode. 0: 16-bit data correspond with LCD_D[15:0] 1: 16-bit data correspond with LCD_D[17:10], LCD_D[8:1]	RW												
30:28	BST	Burst Length Selection. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">Burst Length</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>4 word</td> </tr> <tr> <td>001</td> <td>8 word</td> </tr> <tr> <td>010</td> <td>16 word</td> </tr> <tr> <td>011</td> <td>32 word</td> </tr> <tr> <td>100</td> <td>64 word</td> </tr> </tbody> </table>	Burst Length		000	4 word	001	8 word	010	16 word	011	32 word	100	64 word	RW
Burst Length															
000	4 word														
001	8 word														
010	16 word														
011	32 word														
100	64 word														
27	OUTRGB	Bpp16 RGB output mode. 0: RGB565; 1: RGB555. When 16-bit panel is used, this bit is valid	RW												
26	OFUP	Output FIFO under run protection. 0: disable; 1: enable.	RW												
23:16		Keep these bits to 0													
15		keep this bit to 0.													
14	DACTE	DAC loop back test.	RW												
13	EOFM	Mask end of frame interrupt. 0: INT-disabled; 1: INT-enabled.	RW												
12	SOFM	Mask start of frame interrupt. 0: INT-disabled; 1: INT-enabled.	RW												
11	OFUM	Mask out FIFO under run interrupt. 0: INT-disabled; 1: INT-enabled.	RW												
10	IFUM0	Mask in FIFO 0 under run interrupt. 0: INT-disabled; 1: INT-enabled.	RW												
9	IFUM1	Mask in FIFO 1 under run interrupt. 0: INT-disabled; 1: INT-enabled.	RW												
8	LDDM	Mask LCD disable done interrupt. 0: INT-disabled; 1: INT-enabled.	RW												
7	QDM	Mask LCD quick disable done interrupt. 0: INT-disabled; 1: INT-enabled.	RW												
6	BEDN	Endian selection. 0: same as system Endian; 1: reverse endian format.	RW												
5	PEDN	Endian in byte. 0: msb first; 1: lsb first.	RW												
4	DIS	Disable controller indicate bit. 0: enable; 1: in disabling or disabled.	RW												
3	ENA	Enable controller. 0: disable; 1: enable.	W												
2:0	BPP0	Bits Per Pixel of foreground0. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">Bits Per Pixel</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>1bpp</td> </tr> <tr> <td>001</td> <td>2bpp</td> </tr> </tbody> </table>	Bits Per Pixel		000	1bpp	001	2bpp	R						
Bits Per Pixel															
000	1bpp														
001	2bpp														

			010	4bpp	
			011	8bpp	
			100	15/16bpp	
			101	18bpp/24bpp	
			110	24bpp compressed	
			111	30bpp	

5.6.3 Status Register (LCDSTATE)

	LCDSTATE																0x13050034, 0x130a0034															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								QD	Reserved	EOF	SOF	OUF	IFU0	IFU1	LDD
RST	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect, read as zero.	R
7	QD	LCD Quick disable. 0: not been quick disabled; 1: quick disabled done.	RW
6	Reserved	Writing has no effect, read as zero.	R
5	EOF	End of Frame indicate bit.	RW
4	SOF	Start of Frame indicate bit.	RW
3	OUF	Out FIFO under run.	RW
2	IFU0	In FIFO 0 under run.	RW
1	IFU1	In FIFO 1 under run.	RW
0	LDD	LCD disable. 0: not been normal disabled; 1: been normal disabled.	RW

5.6.4 OSD Configure Register (LCDOSDC)

	LCDOSDC																0x13050100, 0x130a0100																								
Bit																	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved																PREMULTI1	COEF_SEL1	PREMULTI0	COEF_SELO	ALPHAMD1		SOFM1	EOFM1	Reserved	OSDIV	SOFM0	EOFM0	DENDM	Reserved					F1EN	F0EN	ALPHAEN	ALPHAMD0	OSDEN		
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
23	PREMULTI1	Premulti enable of foreground1 : 0 : data has been premulti and don't need premulti 1 : data should be premulti by lcd	R

22:21	COEF_SLE1	Select coefficient for foreground1 : 00 : 0 01 : 1 10 : alpha0 11 : 1-alpha0	R
20	PREMULTI0	Premulti enable of foreground0 : 0 : data has been premulti and don't need premulti 1 : data should be premulti by lcd	R
19:18	COEF_SLE0	Select coefficient for foreground0 : 00 : 0 01 : 1 10 : alpha1 11 : 1-alpha1	R
17	ALPHAMD1	Alpha blending mode for foreground1. 0: One transparency for the whole graphic, and the LCDALPHA register is used for transparency 1: One transparency for each pixel in one graphic, and the alpha value is coming from each pixel data	R
15	SOFM1	Start of frame interrupt mask for foreground 1.	RW
14	EOFM1	End of frame interrupt mask for foreground 1.	RW
13	Reserved	Writing has no effect, read as zero.	R
11	SOFM0	Start of frame interrupt mask for foreground 0.	RW
10	EOFM0	End of frame interrupt mask for foreground 0.	RW
9	DENDM	Display end interrupt mask	RW
8:5	Reserved	Writing has no effect, read as zero.	R
4	F1EN	1: Foreground 1 is enabled 0: Foreground 1 is disabled	R
3	F0EN	1: Foreground 0 is enabled 0: Foreground 0 is disabled.	R
2	ALPHAEN	1: Alpha blending is enabled 0: Alpha blending is disabled	RW
1	ALPHAMD0	Alpha blending mode for foreground0. 0: One transparency for the whole graphic, and the LCDALPHA register is used for transparency 1: One transparency for each pixel in one graphic, and the alpha value is coming from each pixel data	R
0	OSDEN	OSD mod enable. 1: enabled. And you can use F0 F1 0: disabled Set this bit to 1	RW

5.6.5 OSD Control Register (LCDOSDCTRL)

LCDOSDCTRL															0x13050104, 0x130a0104																
Bit															15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
															ipu_clken																
RST															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW																		
15	IPU_CLKEN	IPU clock enable : 1 : enable; 0: disable	RW																		
5	RGB0	Bpp16 RGB mode of foreground0. 0: RGB565; 1: RGB555.	R																		
4	RGB1	Bpp16 RGB mode of foreground1. 0: RGB565; 1: RGB555.	R																		
2:0	BPP1	Bits Per Pixel of OSD channel 1. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">Bits Per Pixel</th> </tr> </thead> <tbody> <tr><td>000</td><td>Reserved</td></tr> <tr><td>001</td><td>Reserved</td></tr> <tr><td>010</td><td>Reserved</td></tr> <tr><td>011</td><td>Reserved</td></tr> <tr><td>100</td><td>15/16bpp</td></tr> <tr><td>101</td><td>18bpp/24bpp</td></tr> <tr><td>110</td><td>24bpp compressed</td></tr> <tr><td>111</td><td>30bpp</td></tr> </tbody> </table>	Bits Per Pixel		000	Reserved	001	Reserved	010	Reserved	011	Reserved	100	15/16bpp	101	18bpp/24bpp	110	24bpp compressed	111	30bpp	R
Bits Per Pixel																					
000	Reserved																				
001	Reserved																				
010	Reserved																				
011	Reserved																				
100	15/16bpp																				
101	18bpp/24bpp																				
110	24bpp compressed																				
111	30bpp																				

5.6.6 OSD State Register (LCDOSDS)

LCDOSDS															0x13050108, 0x130a0108																
Bit															15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
															SOF1	EOF1	Reserved	SOF0	EOF0	Reserved	DEND	Reserved									
RST															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	SOF1	Start of frame flag for foreground 1.	RW
14	EOF1	End of frame flag for foreground 1.	RW
13:12	Reserved	Writing has no effect, read as zero.	R
11	SOF0	Start of frame flag for foreground 0.	RW
10	EOF0	End of frame flag for foreground 0.	RW
9	Reserved	Writing has no effect, read as zero.	R

8	DEND	Display end, 1: display end; 0 : not end	R
7:0	Reserved	Writing has no effect, read as zero.	R

5.6.7 Background0 Color Register (LCDBGC0)

LCDBGC0																0x1305010C, 0x130A010C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								Red [7:0]								Green [7:0]								Blue [7:0]							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:27	Reserved	Writing has no effect, read as zero.	R
23:16	Red	Red part or Y part of background0.	RW
15:8	Green	Green part or Cb part of background0.	RW
7:0	Blue	Blue part or Cr part of background0.	RW

5.6.8 Background1 Color Register (LCDBGC1)

LCDBGC1																0x130502C4,0x130A02C4																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								Red [7:0]								Green [7:0]								Blue [7:0]							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:27	Reserved	Writing has no effect, read as zero.	R
23:16	Red	Red part or Y part of background1.	RW
15:8	Green	Green part or Cb part of background1.	RW
7:0	Blue	Blue part or Cr part of background1.	RW

5.6.9 Foreground Color Key Register 0 (LCDKEY0)

LCDKEY0																0x13050110,0x130a0110																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	KEYEN	KEYMD	Reserved								Red [7:0]								Green [7:0]								Blue [7:0]							
RST			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW
31	KEYEN	The enable bit of color key for foreground 0.	RW
30	KEYMD	Color key mod. 0: color key; 1: mask color key.	RW
29:27	Reserved	Writing has no effect, read as zero.	R
23:16	Red	Red part of color key for foreground 0.	RW
15:8	Green	Green part of color key for foreground 0.	RW
7:0	Blue	Blue part of color key for foreground 0.	RW

5.6.10 Foreground Color Key Register 1 (LCDKEY1)

LCDKEY1																0x13050114,0x130a0114																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEYEN	KEYMD	Reserved						Red [7:0]				Green [7:0]				Blue [7:0]															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	KEYEN	The enable bit of color key for foreground 1.	RW
30	KEYMD	Color key mod. 0: color key; 1: mask color key.	RW
29:27	Reserved	Writing has no effect, read as zero.	R
23:16	Red	Red part of color key for foreground 1.	RW
15:8	Green	Green part of color key for foreground 1.	RW
7:0	Blue	Blue part of color key for foreground 1.	RW

5.6.11 ALPHA Register (LCDALPHA)

LCDALPHA																0x13050118,0x130a0118																
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																ALPHA 0[7:0]				ALPHA 0[7:0]												
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:8	ALPHA1	The alpha value of foreground1 for one graphic with one transparency.	R
7:0	ALPHA0	The alpha value of foreground0 for one graphic with one transparency.	R

The formula of alpha blending is as follows:

$$PIXEL_{0/1} = [Alpha_{0/1} * Foreground_{0/1}]$$

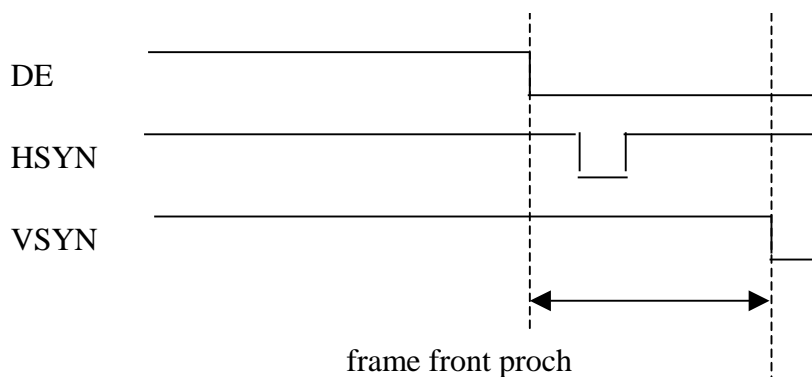
$$F_{0/1} = [0,1, Alpha_{1/0}, (1 - Alpha_{1/0})]$$

$$NewPixel = [F_0 * PIXEL0 + F_1 * PIXEL1]$$

5.6.12 IPU Restart (LCDIPUR)

LCDIPUR																0x1305011C,0x130A011C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IPUREN		Reserved								IPUR																					
RST			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	IPUREN	IPU restart function enable. 0:disable; 1:enable.	RW
30:24	Reserved	Writing has no effect, read as zero.	R
23:0	IPUR	This register is indicating when one frame is end, how long the panel can wait for the next frame data from IPU. In common, set this number larger than frame front porch and near to ((HT-0) X (VPE-VPS))/3. This signal only use when foreground1 work in IPU mode. Trigger IPU transfer the last frame again to avoid output FIFO under run.	RW



5.6.13 RGB Control (LCDRGBC)

LCDRGBC															0x13050090,0x130A0090																
Bit															15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
															RGBDM	DMM	Reserved					422	RGBFMT	OddRGB		Reserved	EvenRGB				
RST															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW																		
15	RGBDM	<p>RGB with dummy data enable.</p> <p>Only useful for RGB serial mode. If this bit set to 1, the one pixel include 4 clock periods, that Red, Green, Blue and Dummy data. Dummy is equal to 0.</p> <p>0: Disable; 1: Enable.</p>	RW																		
14	DMM	<p>RGB dummy mode.</p> <p>0: R-G-B-Dummy</p> <p>1: Dummy-R-G-B</p>																			
13:9	Reserved	Writing has no effect, read as zero.	R																		
8	422	<p>Change 444 to 422.</p> <p>0: not change; 1: change to YUV.</p> <p>This bit only use in OSD mode. Change 444 data to 422 and sent to TV encoder.</p> <p>Please notice that the data will be translated as 16 bits parallel. And only half of it will be transfer. (YCb or YCr in one pixel). If you not use OSD mode and TV encoder, please set this bit to 0.</p> <p>When use this function with IPU transfer data to an interlaced TV, please set IPU output as RGB 888, and OSDBPP to 24. or IPU output data as PACKAGE(YCbYCr) and OSDBPP to 16.</p>	RW																		
7	RGBFMT	RGB format enable : 0: disable; 1: enable.	RW																		
6:4	OddRGB	<p>Odd line serial RGB data arrangement, useful for RGB serial mode only. *Please notice that you must set 000 when use 16/18parallel mode.</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">RGB mode</th> </tr> </thead> <tbody> <tr><td>000</td><td>RGB</td></tr> <tr><td>001</td><td>RBG</td></tr> <tr><td>010</td><td>GRB</td></tr> <tr><td>011</td><td>GBR</td></tr> <tr><td>100</td><td>BRG</td></tr> <tr><td>101</td><td>BGR</td></tr> <tr><td>110</td><td>Reserved</td></tr> <tr><td>111</td><td>Reserved</td></tr> </tbody> </table>	RGB mode		000	RGB	001	RBG	010	GRB	011	GBR	100	BRG	101	BGR	110	Reserved	111	Reserved	RW
RGB mode																					
000	RGB																				
001	RBG																				
010	GRB																				
011	GBR																				
100	BRG																				
101	BGR																				
110	Reserved																				
111	Reserved																				
3	Reserved	Writing has no effect, read as zero.	R																		
2:0	EvenRGB	Even line serial RGB data arrangement, useful for RGB serial mode	RW																		

		only. *Please notice that you must set 000 when use 16/18parallel mode.																			
		<table border="1"> <thead> <tr> <th colspan="2">RGB mode</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>RGB</td> </tr> <tr> <td>001</td> <td>RBG</td> </tr> <tr> <td>010</td> <td>GRB</td> </tr> <tr> <td>011</td> <td>GBR</td> </tr> <tr> <td>100</td> <td>BRG</td> </tr> <tr> <td>101</td> <td>BGR</td> </tr> <tr> <td>110</td> <td>Reserved</td> </tr> <tr> <td>111</td> <td>Reserved</td> </tr> </tbody> </table>	RGB mode		000	RGB	001	RBG	010	GRB	011	GBR	100	BRG	101	BGR	110	Reserved	111	Reserved	
RGB mode																					
000	RGB																				
001	RBG																				
010	GRB																				
011	GBR																				
100	BRG																				
101	BGR																				
110	Reserved																				
111	Reserved																				

5.6.14 Virtual Area Setting (LCDVAT)

	LCDVAT																0x1305000C,0x130A000C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				HT												Reserved				VT											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	HT	Horizontal Total size. (in dot clock, sum of display area and blank space)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	VT	Vertical Total size. (in line clock, sum of display area and blank space)	RW

5.6.15 Display Area Horizontal Start/End Point (LCDDAH)

	LCDDAH																0x13050010,0x130a0010															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				HDS												Reserved				HDE											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	HDS	Horizontal display area start. (in dot clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	HDE	Horizontal display area end. (in dot clock)	RW

5.6.16 Display Area Vertical Start/End Point (LCDDAV)

LCDDAV																0x13050014,0x130a0014																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved				VDS												Reserved				VDE												
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	VDS	Vertical display area start position. (in line clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	VDE	Vertical display area end position. (in line clock)	RW

5.6.17 Foreground 0 XY Position Register (LCDXYP0)

LCDXYP0																0x13050120,0x130a0120																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved				YPOS												Reserved				XPOS												
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	YPOS	The Y position of top-left part for foreground 0.	R
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	XPOS	The X position of top-left part for foreground 0.	R

5.6.18 Foreground 1 XY Position Register (LCDXYP1)

LCDXYP1																0x13050124,0x130a0124																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved				YPOS												Reserved				XPOS												
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	YPOS	The Y position of top-left part for foreground 1.	R

15:12	Reserved	Writing has no effect, read as zero.	R
11:0	XPOS	The X position of top-left part for foreground 1.	R

5.6.19 Foreground 0 Size Register (LCDSIZE0)

LCDSIZE0																0x13050128,0x130a0128																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				Height												Reserved				Width											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	Height	The height-1 of foreground 0.	R
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	Width	The width-1 of foreground 0.	R

When use TVE interlaced mode, please set the area of F0 and F1 aligned with BST.

5.6.20 Foreground 1 Size Register (LCDSIZE1)

LCDSIZE1																0x1305012c,0x130a012c																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				Height												Reserved				Width											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	Height	The height-1 of foreground 1.	R
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	Width	The width-1 of foreground 1.	R

5.6.21 Vertical Synchronize Register (LCDVSYNC)

LCDVSYNC																0x13050004,0x130a0004																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				VPS												Reserved				VPE											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	VPS	V-Sync Pulse start position, fixed to 0. (in line clock)	R
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	VPE	V-Sync Pulse end position. (in line clock)	RW

5.6.22 Horizontal Synchronize Register (LCDHSYNC)

LCDHSYNC																0x13050008,0x130a0008																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				HPS												Reserved				HPE											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	HPS	H-Sync pulse start position. (in dot clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	HPE	H-Sync pulse end position. (in dot clock)	RW

5.6.23 PS Signal Setting (LCDPS)

LCDPS																0x13050018,0x130a0018																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				PSS												Reserved				PSE											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	PSS	PS signal start position. (in dot clock) In STN mode, PS signal is ignored. But this register is used to define the AC BIAS signal. AC BIAS signal will toggle very N lines per frame. PSS defines the Toggle position.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	PSE	PS signal end position. (in dot clock) In STN mode, PSE defines N, which described in PSS.	RW

5.6.24 CLS Signal Setting (LCDCLS)

LCDCLS																0x1305001C,0x130a001C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				CLSS												Reserved				CLSE											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	CLSS	CLS signal start position. (in dot clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	CLSE	CLS signal end position. (in dot clock)	RW

5.6.25 SPL Signal Setting (LCDSPL)

LCDSPL																0x13050020,0x130a0020																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				SPLS												Reserved				SPLE											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	SPLS	SPL signal start position. (in dot clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	SPLE	SPL signal end position. (in dot clock)	RW

* In test mode this register use to keep TV encoder module's output data: comp_luma([25:16]) and chroma([9:0]).

5.6.26 REV Signal Setting (LCDREV)

LCDREV																0x13050024,0x130a0024																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				REVS												Reserved															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	REVS	REV signal start position. (in dot clock)	RW
15:0	Reserved	Writing has no effect, read as zero.	R

5.6.27 Interrupt ID Register (LCDIID)

LCDIID is a read-only register that contains a copy of the Frame ID register (LCDFID) from the descriptor currently being processed when a start of frame (SOF) or end of frame (EOF) interrupt is generated. LCDIID is written to only when an unmasked interrupt of the above type is signaled and there are no other unmasked interrupts in the LCD controller pending. As such, the register is considered to be sticky and will be overwritten only when the signaled interrupt is cleared by writing the LCD controller status register. For dual-panel displays, LCDIID is written only when both channels have reached a given state.

LCDIID is written with the last channel to reach that state. (i.e. LCDFID of the last channel to reach SOF would be written in LCDIID if SOF interrupts are enabled). Reserved bits must be written with zeros and reads from them must be ignored.

LCDIID																0x13050038,0x130a0038																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	IID																																
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	IID	A copy of Frame ID register, which transferred from Descriptor.	RW

5.6.28 Descriptor Address Registers (LCDDAx)

A frame descriptor is a 4-word block, aligned on 4-word (16-byte) boundary, in external memory:

WORD [0] contains the physical address for next LCDDAx.

WORD [1] contains the physical address for LCDSAx.

WORD [2] contains the value for LCDFIDx.

WORD [3] contains the value for LCDCMDx.

Software must write the physical address of the first descriptor to LCDDAx before enabling the LCD Controller. Once the LCD Controller is enabled, the first descriptor is read, and all 4 registers are written by the DMAC. The next frame descriptor pointed to by LCDDAx is loaded into the registers for the associated DMA channel after all data for the current descriptor has been transferred.

NOTE: If only one frame buffer is used in external memory, the LCDDAx field (word [0] of the frame descriptor) must point back to itself. That is to say, the value of LCDDAx is the physical

address of itself.

Read/write registers LCDDA0 and LCDDA1, corresponding to DMA channels 0 and 1, contain the physical address of the next descriptor in external memory. The DMAC fetches the descriptor at this location after finishing the current descriptor. On reset, the bits in this register are zero. The target address must be aligned to 16-byte boundary. Bits [3:0] of the address must be zero.

LCDDA0, LCDDA1		0x13050040, 0x13050050, 0x130a0040, 0x130a0050																															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DA0, 1																																	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	DA0, 1	Next descriptor physical address. And descriptor structure as following: WORD [0]: next descriptor physical address WORD [1]: the buffer physical address WORD [2]: the buffer ID value (Only for debug) WORD [3]: the buffer property. The value is same as LCDCMD	RW

5.6.29 Source Address Registers (LCDSA)

Registers LCDSA0 and LCDSA1, corresponding to DMA channels 0 and 1, contain the physical address of frame buffer in external memory. The address must be aligned on a 4, 8, or 16 word boundary according to register LCDCTRL.BST. If this descriptor is for frame data, LCDSAx points to the memory location of the frame buffer. This address is incremented by hardware as the DMAC fetches data from memory. If desired, the Frame ID Register can be used to hold the initial frame source address.

LCDSA0, LCDSA1		0x13050044, 0x13050054, 0x130a0044, 0x130a0054																														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SA0, 1																																
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	SA0, 1	Buffer start address. (Only for driver debug)	R

5.6.30 Frame ID Registers (LCDFIDx)

Registers LCDFID0 and LCDFID1, corresponding to DMA channels 0 and 1, contain an ID field that

describes the current frame. The particular use of this field is up to the software. This ID register is copied to the LCD Controller Interrupt ID Register when an interrupt occurs.

	LCDFID0, LCDFID1																0x13050048, 0x13050058, 0x130a0048, 0x130a0058																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	FID0, 1																																		
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	FID0, 1	Frame ID. (Only for debug)	R

5.6.31 DMA Command Registers (LDCMDx)

	LDCMD0, LDCMD1																0x1305004C, 0x1305005C, 0x130A004C, 0x130A005C																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
	SOFINT	EOFINT	CMD	COMPEN	FRM_EN	FIELD_SEL	16x16BLOCK	LEN																												
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Bits	Name	Description	RW
31	SOFINT	Enable start of frame interrupt. When SOFINT =1, the DMAC sets the start of frame bit (LCDSTATE.SOF) when starting a new frame. The SOF bit is set after a new descriptor is loaded from memory and before the frame data is fetched. In dual-panel mode, LCDSTATE.SOF is set only when both channels reach the start of frame and both frame descriptors have SOFINT set.	R
30	EOFINT	Enable end of frame interrupt. When EOFINT =1, the DMAC sets the end of frame bit (LCDSTATE.EOF) after fetching the last word in the frame buffer. In dual-panel mode, LCDSTATE.EOF is set only when both channels reach the end of frame and both frame descriptors have EOFINT set.	R
29	CMD	It is used to distinguish command and data in lcm mode. And it is only loaded via DMA channel 0. 1: The data is command 0: The data is data	R

28		Keep this bit to 0.	
27	COMPE N	It indicate this frm is 16/24 bpp compressed or not. 0: not compressed; 1: compressed.	R
26	FRM_EN	Indicat this frm is enable. 0:disable; 1:enable.	R
25	FIELD_S EL	Field select for interlace : 0:odd field or no interlace; 1: even field	
24	16x16BLO CK	Fetch data by 16x16 block,0:disable;1:enable.	R
23:0	LEN	The buffer length value. (in WORD) The LEN bit field determines the number of bytes of the buffer size pointed by LCDSAx. LEN = 0 is not valid. DMAC fetch data according to LEN. Each time one or more word(s) been fetched, LEN is decreased automatically. Software can read LEN. <i>*When you use decompressed function, the LEN should be the line number, not word number.</i> <i>*When you use 16x16block mode, the LEN should be the line number, not word number.</i>	R

5.6.32 DMA OFFSIZE Registers (LCDOFFSx)

	LCDOFFS0, LCDOFFS1																0x13050060, 0x13050070, 0x130a0060, 0x130a0070																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
	Reserved								OFFSIZE																											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

Bits	Name	Description	RW
23:0	OFFSIZE0, 1	OFFSIZE value for DMA 0,1. Indicate the offset in word. <i>*please notice that when you need OFFSIZE function, to set this reg to an un-zero value and also need to set LCDPW0, 1 to indicate how much word in one line of this frame.</i> <i>*When you use decompress function, you must use this to indicate how many word between each of line.</i> <i>*When you use 16x16block mode, you must use this to indicate how many word between each of block</i>	R

5.6.33 DMA Page Width Registers (LCDPWx)

	LCDPW0, LCDPW1																0x13050064, 0x13050074, 0x130a0064, 0x130a0074															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								PAGEWIDTH																							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
23:0	PAGEWIDTH0, 1	Page width for DMA 0,1. <i>* When you set LCDOFFS.OFFSIZE0/1 to 0, you need keep the PAGEWIDTH0/1 0.</i> <i>*When you use decompress function, you don't need set this register, dma will get pagewidth of every line automatically.</i> <i>*When you use 16x16block mode, you use this to indicate how many words per line.</i>	R

5.6.34 DMA Command Counter Registers (LCDCNUMx)

When LCDCMD.COMD = 1, 0x13050068, 0x13050078 is use as LCDCNUM0, 1 are not used now, set it to 0.

	LCDCNUM0, LCDCNUM1																0x13050068, 0x13050078, 0x130a0068, 0x130a0078															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																CNUM															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:0	CNUM0,1	Commands' number in this frame transfer by DMA. (only use in Smart LCD mode)	R

5.6.35 DMA Command Counter Registers (LCDPOSx)

When LCDCMD.COMD = 0, 0x13050068, 0x13050078 is use as LCDPOS0, 1.

LCDPOS0, LCDPOS1																0x13050068, 0x13050078, 0x130a0068, 0x130a0078																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	ALPHAMD1	RGB0/1	BPP0/1			PREMULTI0/1	COEF_SLE0/1	YPOS										XPOS															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW																		
31	ALPHAMD1	Alpha blending mode for foreground0/1. 0: One transparency for the whole graphic, and the LCDALPHA register is used for transparency 1: One transparency for each pixel in one graphic, and the alpha value is coming from each pixel data	R																		
30	RGB0/1	Bpp16 RGB mode of foreground0/1. 0: RGB565; 1: RGB555.	R																		
29:27	BPP0/1	Bits Per Pixel of OSD channel 1. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">Bits Per Pixel</th> </tr> </thead> <tbody> <tr><td>000</td><td>Reserved</td></tr> <tr><td>001</td><td>Reserved</td></tr> <tr><td>010</td><td>Reserved</td></tr> <tr><td>011</td><td>Reserved</td></tr> <tr><td>100</td><td>15/16bpp</td></tr> <tr><td>101</td><td>18bpp/24bpp</td></tr> <tr><td>110</td><td>24bpp compressed</td></tr> <tr><td>111</td><td>30bpp</td></tr> </tbody> </table>	Bits Per Pixel		000	Reserved	001	Reserved	010	Reserved	011	Reserved	100	15/16bpp	101	18bpp/24bpp	110	24bpp compressed	111	30bpp	R
Bits Per Pixel																					
000	Reserved																				
001	Reserved																				
010	Reserved																				
011	Reserved																				
100	15/16bpp																				
101	18bpp/24bpp																				
110	24bpp compressed																				
111	30bpp																				
26	PREMULTI0/1	Premulti enable of foreground0/1 : 0 : data has been premultiplied and don't need premulti 1 : data should be premultiplied by lcd	R																		
25:24	COEF_SLE0/1	Select coefficient for foreground0/1 : 00 : 0 01 : 1 10 : alpha1 11 : 1-alpha1	R																		
23:12	YPOS0,1	The Y position of top-left part for foreground 0/1	R																		
11:0	XPOS0,1	The X position of top-left part for foreground 0/1	R																		

5.6.36 Foreground x Size in Descriptor (LCDDESSIZEx)

When LCDCMD.CMD = 0, 0x1305006C, 0x1305007C is use as LCDDESSIZE0, 1, to indicator the next frame foreground0, 1's size.

LCDESSIZE0, 1																0x1305006C, 0x1305007C, 0x130A006C, 0x130A007C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALPHA								Height								Width															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:24	ALPHA	The alpha value of foreground0/1 for one graphic with one transparency.	R
23:12	Height	The height-1 of foreground 0/1.	R
11:0	Width	The width-1 of foreground 0/1.	R

5.6.37 Priority level threshold configure Register (LCDPCFG)

LCDPCFG																0x130502C0, 0x130A02C0																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Lcd_pri_md	HP_BST			Reserved	Pcfg2								Pcfg1						Pcfg0												
RST		0	1	1		1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The 3 thresholds cut the output fifo into 4 space,. When the entries remained triggers one of them, the priority level will be altered by hardware. And the 3 thresholds must follow the order: pcfg2 ≥ pcfg1 ≥ pcfg0.

Priority level

Priority_Iv	Space
11	Empty ~ threshold0
10	Threshold0 ~ threshold1
01	Threshold1 ~ threshold2
00	Threshold2 ~ full

Bits	Name	Description	RW
31	Lcd_pri_md	Lcd priority mode. 0: use lcd dynamic priority level; 1: use arbiter priority level.	RW
30:28	HP_BST	Highest priority Burst Length Selection. Burst Length	RW

		000	4 word	
		001	8 word	
		010	16 word	
		011	32 word	
		101	Contiue16	
		100	64 word	
		111	disable	
27	Reserved	Writing has no effect, read as zero.		R
26:18	Pcfg2	Threshold2: 0~511.		RW
17:9	Pcfg1	Threshold1: 0~511.		RW
8:0	Pcfg0	Threshold0: 0~511		RW

5.6.38 Dual LCDC Channel Control(LCDCDUALCTRL)

	LCDCDUALCTRL																0x130502C8,0x130A02C8																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved																ipu_wr_sel(lcdc1)	Reserved	Tft_sel(lcdc1)	Pri_ipu_en	Pri_ipu	Pri_lcd_en	Pri_lcd										
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW
8	ipu_wr_sel(lcdc1)	Select which IPU is able to write back, this field is just available in lcdc1 : 0:ipu1; 1:ipu0	RW
6	Tft_sel(lcdc1)	Select which controller output to the tft/slcd panel, this field is just available in lcdc1 : 0:lcdc1; 1:lcdc0	RW
5	Pri_ipu_en	1:fix the priority of ipu0/1 in lcd internal arbiter; 0: use priority of ipu0/1 generated by lcd in lcd internal arbiter	RW
3:4	Pri_ipu	Set ipu0/1 priority, just active when pri_ipu_en is 1	RW
2	Pri_lcd_en	1:fix the priority of lcdc0/1 in lcd internal arbiter; 0: use priority of lcdc0/1 generated by lcd in lcd internal arbiter	RW
1:0	Pri_lcd	Set lcdc0/1 priority, just active when pri_lcd_en is 1	RW

5.6.39 Image Enhancement CFG (LCDENH_CFG)

LCDENH_CFG	0x13050400,0x130a0400
-------------------	------------------------------

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																							DITHER_EN	YCC2RGB_EN	SATURATION_EN	VEE_EN	HUE_EN	BRIGHTNESS_EN	CONTRAST_EN	RGB2YCC_EN	GAMMA_EN	ENH_EN
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
9	DITHER_EN	Dither enable : 0 : disable 1: enable	RW
8	YCC2RGB_EN	YCbCr to RGB enable : 0 : disable 1: enable	RW
7	SATURATION_EN	Saturation enable : 0 : disable 1: enable	RW
6	VEE_EN	Visibility enhance enable : 0: disable 1: enable	RW
5	HUE_EN	Hue enable : 0 : disable 1 : enable	RW
4	BRIGHTNESS_EN	Brightness enable : 0 : disable 1 : enable	RW
3	CONTRAST_EN	Contrast enable : 0 : disable 1 : enable.	RW
2	RGB2YCC_EN	RGB to YCbCr enable : 0 : disable 1 : enable	RW
1	GAMMA_EN	Gamma control enable : 0 : disable 1: enable	RW
0	ENH_EN	Enhancement enable: 0: disable 1: enable	RW

5.6.40 Color Space Conversion CFG (LCDENH_CSCCFG)

LCDENH_CSCCFG																0x13050404,0x130a0404																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW			
3:2	YCC2RGBMD	00 : 601WIDE; 01 : 601NARROW; 10 : 709WIDE; 11 : 709NARROW	RW			
		<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 100px;"></td> <td style="text-align: center;">609</td> <td style="text-align: center;">701</td> </tr> </table>		609	701	
	609	701				

		<table border="1"> <tr> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">WIDE</td> <td> R= $Y+1.371(Cr-128)$ G= $Y-0.689(Cr-128)-0.336(Cb-128)$ B= $Y+1.732(Cb-128)$ </td> <td> R= $Y+1.540(Cr-128)$ G= $Y-0.459(Cr-128)-0.183(Cb-128)$ B= $Y+1.816(Cb-128)$ </td> </tr> <tr> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">NARROW</td> <td> R= $1.164(Y-16)+1.596(Cr-128)$ G= $1.164(Y-16)-0.813(Cr-128)-0.391(Cb-128)$ B= $1.164(Y-16)+2.018(Cb-128)$ </td> <td> R= $1.164(Y-16)+1.793(Cr-128)$ G= $1.164(Y-16)-0.534(Cr-128)-0.213(Cb-128)$ B= $1.164(Y-16)+2.115(Cb-128)$ </td> </tr> </table> <p>WIDE : RGB range 16-235 NARROW : RGB range 0-255</p>	WIDE	R= $Y+1.371(Cr-128)$ G= $Y-0.689(Cr-128)-0.336(Cb-128)$ B= $Y+1.732(Cb-128)$	R= $Y+1.540(Cr-128)$ G= $Y-0.459(Cr-128)-0.183(Cb-128)$ B= $Y+1.816(Cb-128)$	NARROW	R= $1.164(Y-16)+1.596(Cr-128)$ G= $1.164(Y-16)-0.813(Cr-128)-0.391(Cb-128)$ B= $1.164(Y-16)+2.018(Cb-128)$	R= $1.164(Y-16)+1.793(Cr-128)$ G= $1.164(Y-16)-0.534(Cr-128)-0.213(Cb-128)$ B= $1.164(Y-16)+2.115(Cb-128)$				
WIDE	R= $Y+1.371(Cr-128)$ G= $Y-0.689(Cr-128)-0.336(Cb-128)$ B= $Y+1.732(Cb-128)$	R= $Y+1.540(Cr-128)$ G= $Y-0.459(Cr-128)-0.183(Cb-128)$ B= $Y+1.816(Cb-128)$										
NARROW	R= $1.164(Y-16)+1.596(Cr-128)$ G= $1.164(Y-16)-0.813(Cr-128)-0.391(Cb-128)$ B= $1.164(Y-16)+2.018(Cb-128)$	R= $1.164(Y-16)+1.793(Cr-128)$ G= $1.164(Y-16)-0.534(Cr-128)-0.213(Cb-128)$ B= $1.164(Y-16)+2.115(Cb-128)$										
1:0	RGB2 YCCM D	<p>00 : 601WIDE; 01 : 601NARROW; 10 : 709WIDE; 11 : 709NARROW</p> <table border="1"> <thead> <tr> <th></th> <th>609</th> <th>701</th> </tr> </thead> <tbody> <tr> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">WIDE</td> <td> Y= $0.299R+0.587G+0.114B$ Cb= $-0.172R-0.339G+0.511B+128$ Cr= $0.511R-0.428G-0.083B+128$ </td> <td> Y= $0.213R+0.715G+0.072B$ Cb= $-0.117R-0.394G+0.511B+128$ Cr= $0.511R-0.464G-0.047B+128$ </td> </tr> <tr> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">NARROW</td> <td> Y= $0.257R+0.504G+0.098B+16$ Cb= $-0.148R-0.291G+0.439B+128$ Cr= $0.439R-0.368G-0.071B+128$ </td> <td> Y= $0.183R+0.614G+0.062B+16$ Cb= $-0.101R-0.338G+0.439B+128$ Cr= $0.439R-0.399G-0.040B+128$ </td> </tr> </tbody> </table> <p>WIDE : RGB range 16-235 NARROW : RGB range 0-255</p>		609	701	WIDE	Y= $0.299R+0.587G+0.114B$ Cb= $-0.172R-0.339G+0.511B+128$ Cr= $0.511R-0.428G-0.083B+128$	Y= $0.213R+0.715G+0.072B$ Cb= $-0.117R-0.394G+0.511B+128$ Cr= $0.511R-0.464G-0.047B+128$	NARROW	Y= $0.257R+0.504G+0.098B+16$ Cb= $-0.148R-0.291G+0.439B+128$ Cr= $0.439R-0.368G-0.071B+128$	Y= $0.183R+0.614G+0.062B+16$ Cb= $-0.101R-0.338G+0.439B+128$ Cr= $0.439R-0.399G-0.040B+128$	RW
	609	701										
WIDE	Y= $0.299R+0.587G+0.114B$ Cb= $-0.172R-0.339G+0.511B+128$ Cr= $0.511R-0.428G-0.083B+128$	Y= $0.213R+0.715G+0.072B$ Cb= $-0.117R-0.394G+0.511B+128$ Cr= $0.511R-0.464G-0.047B+128$										
NARROW	Y= $0.257R+0.504G+0.098B+16$ Cb= $-0.148R-0.291G+0.439B+128$ Cr= $0.439R-0.368G-0.071B+128$	Y= $0.183R+0.614G+0.062B+16$ Cb= $-0.101R-0.338G+0.439B+128$ Cr= $0.439R-0.399G-0.040B+128$										

5.6.41 LUMA CFG (LCDENH_LUMACFG)

LCDENH_LUMACFG																0x13050408,0x130a0408																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRIGHTNESS																CONTRAST															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
26:16	BRIGHTNESS	Brightness value : 0x0~0x7ff means 0.9999~-0.9999	RW
10:0	CONTRAST	Contrast value : 0x0~0x7ff means 0~1.9999	RW

5.6.42 CHROMA0 CFG (LCDENH_CHROCFG0)

LCDENH_CHROCFG0																0x1305040c,0x130a040c																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HUE_SIN																HUE_COS															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
27:16	HUE_SIN	Hue sin value : 0xc00~400 means -1~1	RW
11:0	HUE_COS	Hue cos value : 0xc00~400 means -1~1	RW

5.6.43 CHROMA1 CFG (LCDENH_CHROCFG1)

LCDENH_CHROCFG1																0x13050410,0x130a0410																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SATURATION																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
10:0	SATURATION	Saturation value : 0x0~0x7ff means 0~1.9999	RW

5.6.44 DITHER CFG (LCDENH_DITHERCFG)

LCDENH_DITHERCFG																0x13050414,0x130a0414																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

																									DITHERDM_R		DITHERDM_G		DITHERDM_B	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
5:0	DITHERMD_RED, DITHERMD_GREEN, DITHERMD_BLUE	00: 8bit dither 01: 6bit dither 10: 5bit dither 11: 4bit dither	RW

5.6.45 Enhance status (LCDENH_STATUS)

LCDENH_STATUS																0x13050418,0x130a0418																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																							DITHER_DIS	YCC2RGB_DIS	SATURATION_DIS	VEE_DIS	HUE_DIS	BRIGHTNESS_DIS	CONTRAST_DIS	RGB2YCC_DIS	GAMMA_DIS	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
9	DITHER_DIS	Dither disable done	R
8	YCC2RGB_DIS	YCbCr to RGB disable done	R
7	SATURATION_DIS	Saturation disable done	R
6	VEE_DIS	Visibility enhance disable done	R
5	HUE_DIS	Hue disable done	R
4	BRIGHTNESS_DIS	Brightness disable done	R
3	CONTRAST_DIS	Contrast disable done	R
2	RGB2YCC_DIS	RGB to YCbCr disable done	R
1	GAMMA_DIS	Gamma control disable done	R

5.6.46 GAMMA CFG (LCDENH_GAMMA)

LCDENH_GAMMA																0x13050800~0x13050fff,0x130a0800~0x130a0fff																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

		GAMMA_DATA1										GAMMA_DATA0																					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
25:16	GAMMA_DATA1	Gamma data 1, 3, ..., 1023	RW
9:0	GAMMA_DATA0	Gamma data 0, 2, ..., 1022	RW

5.6.47 VEE CFG (LCDENH_VEE)

	LCDENH_VEE															0x13051000~0x130517ff,0x130a1000~0x130a17ff																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											VEE_DATA1										VEE_DATA0											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
25:16	VEE_DATA1	Vee data 1, 3, ..., 1023	RW
9:0	VEE_DATA0	Vee data 0, 2, ..., 1022	RW

5.7 LCD Controller Pin Mapping

There are several mapping schemes for different LCD panels.

5.7.1 TFT Pin Mapping

Pin	Generic	Generic	Generic	24 bit
-----	---------	---------	---------	--------

	8-bit Serial TFT	18-bit Parallel TFT	16-bit Parallel TFT	Parallel
Lcd_pclk/ Slcd_clk	CLK	CLK	CLK	CLK
Lcd_vsync/Slc d_wr	VSYNC	VSYNC	VSYNC	VSYNC
Lcd_hsync/Slc d_dc	HSYNC	HSYN C	HSYNC	HSYNC
Lcd_de	DE	DE	DE	DE
Lcd_ps	-	-	-	
Lcd_cls	-	-	-	
Lcd_rev	-	-	-	
Lcd_spl	-	-	-	
Lcd_dat17	-	R5		R7
Lcd_dat16	-	R4		R6
Lcd_dat15	-	R3	R4	R5
Lcd_dat14	-	R2	R3	R4
Lcd_dat13	-	R1	R2	R3
Lcd_dat12	-	R0	R1	R2
Lcd_dat11	-	G5	R0	G7
Lcd_dat10	-	G4	G5	G6
Lcd_dat9	-	G3	G4	G5
Lcd_dat8	-	G2	G3	G4
Lcd_dat7	R7/G7/B7	G1	G2	G3
Lcd_dat6	R6/G6/B6	G0	G1	G2
Lcd_dat5	R5/G5/B5	B5	G0	B7
Lcd_dat4	R4/G4/B4	B4	B4	B6
Lcd_dat3	R3/G3/B3	B3	B3	B5
Lcd_dat2	R2/G2/B2	B2	B2	B4
Lcd_dat1	R1/G1/B1	B1	B1	B3
Lcd_dat0	R0/G0/B0	B0	B0	B2
Lcd_lo6_o[5]	0	0	0	R1
Lcd_lo6_o[4]	0	0	0	R0
Lcd_lo6_o[3]	0	0	0	G1
Lcd_lo6_o[2]	0	0	0	G0
Lcd_lo6_o[1]	0	0	0	B1
Lcd_lo6_o[0]	0	0	0	B0

5.7.2 Data mapping to GPIO function.

pin name in LCD	mapping to GPIO function
Lcd_dat17/Slcd_dat17	lcd_r7
Lcd_dat16/Slcd_dat16	lcd_r6
Lcd_dat15/Slcd_dat15	lcd_r5
Lcd_dat14/Slcd_dat14	lcd_r4
Lcd_dat13/Slcd_dat13	lcd_r3
Lcd_dat12/Slcd_dat12	lcd_r2
Lcd_dat11/Slcd_dat11	lcd_g7
Lcd_dat10/Slcd_dat10	lcd_g6
Lcd_dat9/Slcd_dat9	lcd_g5
Lcd_dat8/Slcd_dat8	lcd_g4
Lcd_dat7/Slcd_dat7	lcd_g3
Lcd_dat6/Slcd_dat6	lcd_g2
Lcd_dat5/Slcd_dat5	lcd_b7
Lcd_dat4/Slcd_dat4	lcd_b6
Lcd_dat3/Slcd_dat3	lcd_b5
Lcd_dat2/Slcd_dat2	lcd_b4
Lcd_dat1/Slcd_dat1	lcd_b3
Lcd_dat0/Slcd_dat0	lcd_b2
Lcd_lo6_o[5]	lcd_r1
Lcd_lo6_o[4]	lcd_r0
Lcd_lo6_o[3]	lcd_g1
Lcd_lo6_o[2]	lcd_g0
Lcd_lo6_o[1]	lcd_b1
Lcd_lo6_o[0]	lcd_b0

5.8 Display Timing

5.8.1 General 16-bit and 18-bit TFT Timing

This section shows the general 16-bit and 18-bit TFT LCD timing diagram, the polarity of signal “Vsync”, “Hsync”, and “PCLK” can be programmed correspond to the LCD panel specification.

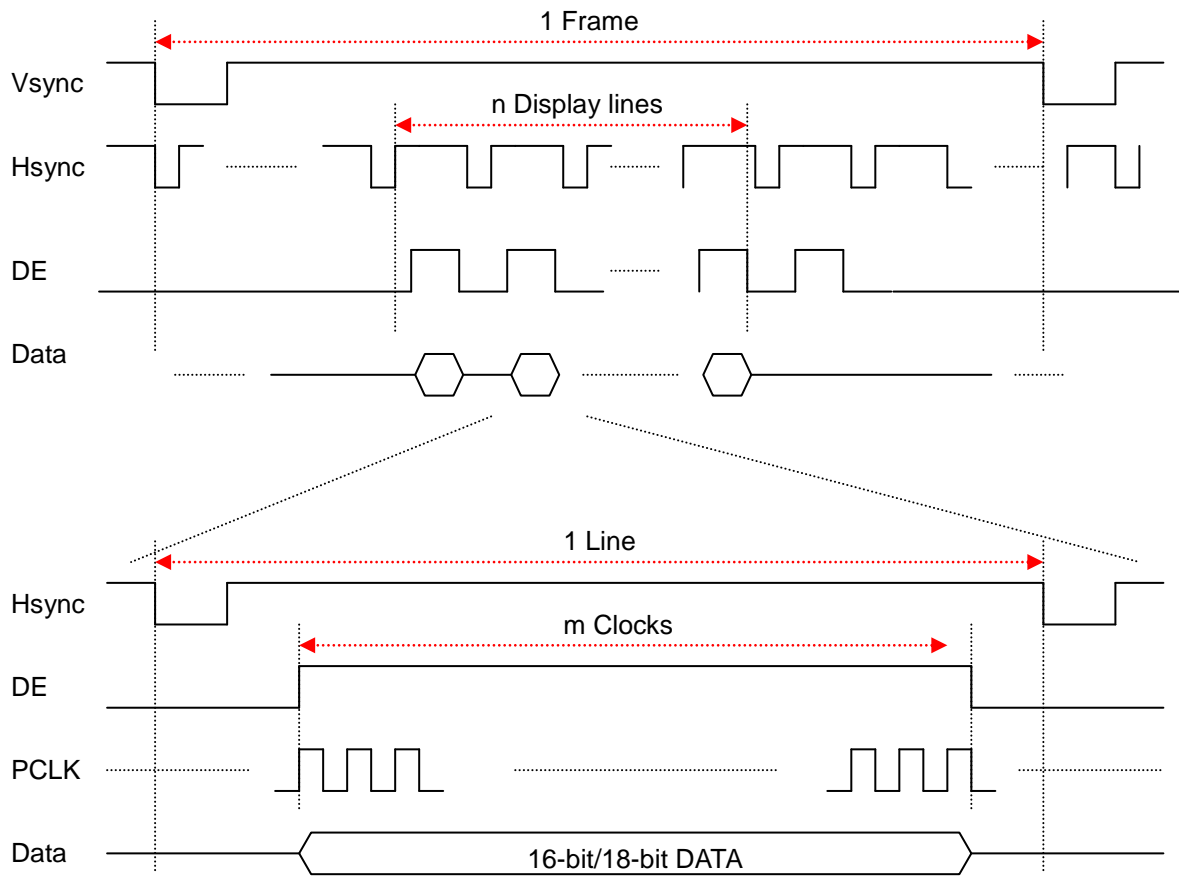


Figure 5-6 General 16-bit and 18-bit TFT LCD Timing

5.8.2 8-bit Serial TFT Timing

This section shows the 8-bit serial TFT LCD timing diagram, the polarity of signal “Vsync”, “Hsync”, and “PCLK” can be programmed correspond to the LCD panel specification.

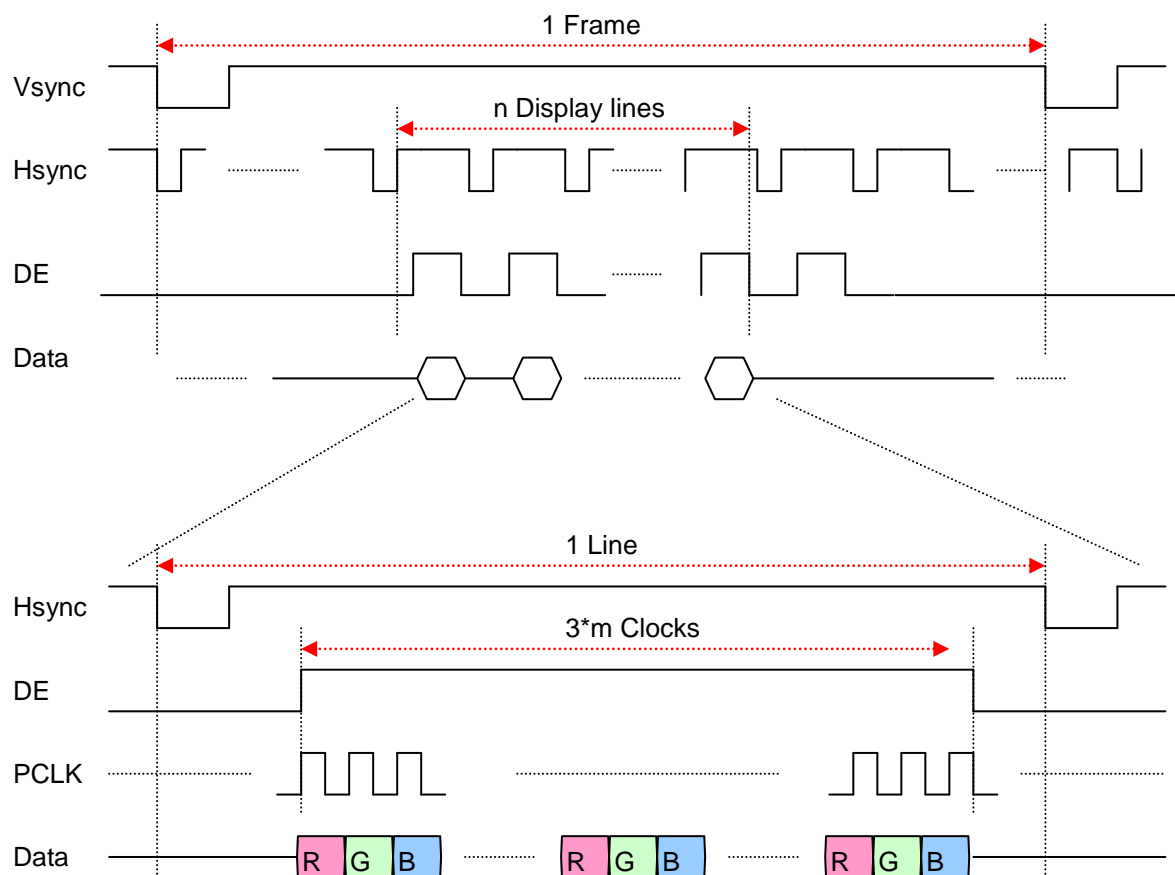


Figure 5-7 8-bit serial TFT LCD Timing (24bpp)

5.8.3 Special TFT Timing

Based on the general TFT LCD support, this controller also provides 4 special signals that can be programmed to general some special timing used for some panel. All 4 signals are worked in two modes: pulse mode and toggle mode. Signal "CLS" is fixed in pulse mode, and "REV" in toggle mode. The work mode of signals "SPL" and "PS" are defined in the special TFT LCD mode 1 to mode 3, either pulse mode or toggle mode. The position and polarity of these 4 signals can be programmed via registers. The Figures show the two modes as follows: (The toggle mode of signal "SPL" is different with the others signal. "SPL" does toggle after display line.)

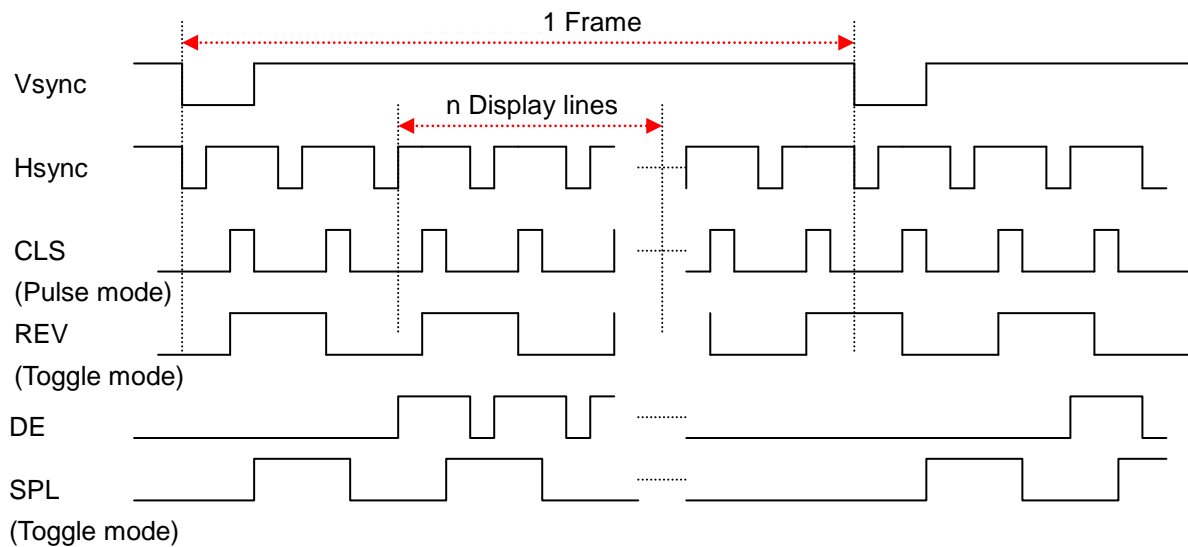


Figure 5-8 Special TFT LCD Timing 1

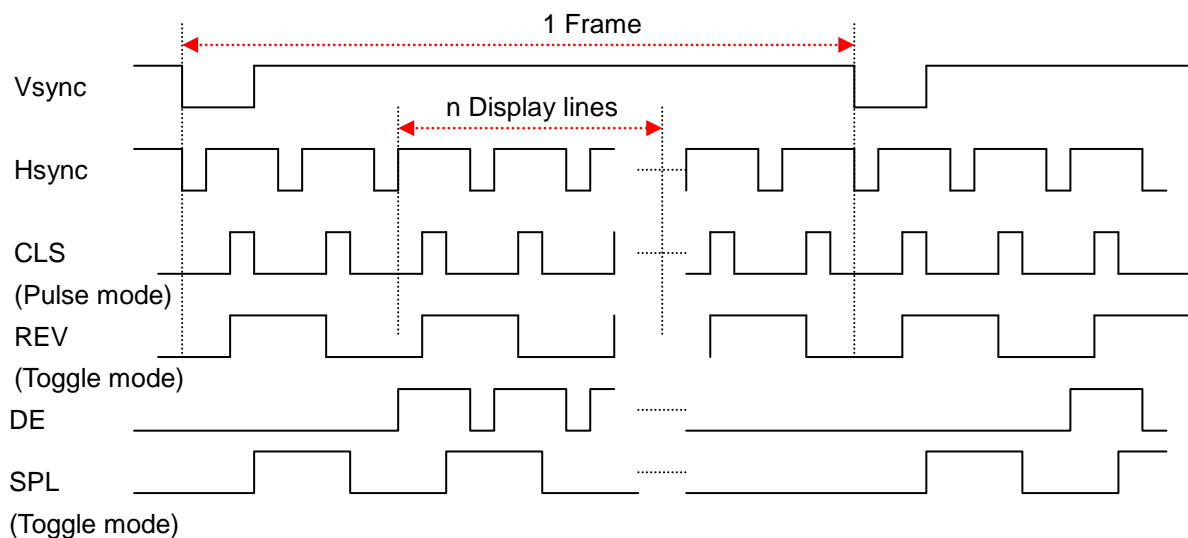


Figure 5-9 Special TFT LCD Timing 2

These two Figures show the timing of pulse mode and toggle mode, the pulse mode timing is same and the toggle mode timing is different. Timing 1 shows the condition when the total lines in 1 frame is odd (the number of display is even and the number of blank is odd), so the phase of REV inverse at the first line of each frame and the phase of SPL dose not inverse at the first line of each frame. Timing 2 shows the condition when the total lines in 1 frame is even (the number of display is even and the number of blank is even), so the phase of REV and SPL dose not inverse at the first line of each frame.

When LCDC is enabled ,there will be a null line to be add before transferring data to LCD panel. So the toggle mode except SPL signal of special 3 TFT mode is when reset level is high,the first valid edge will be rising edge. SPL signal of special 3 TFT mode is when reset level is high,the first valid edge will be

falling edge.

5.8.4 Delta RGB panel timing

This section shows the Delta RGB timing diagram, the polarity of signal “Vsync”, “Hsync”, and “PCLK” can be programmed. And the odd/even line RGB order also can be programmed correspond to the LCD panel specification.

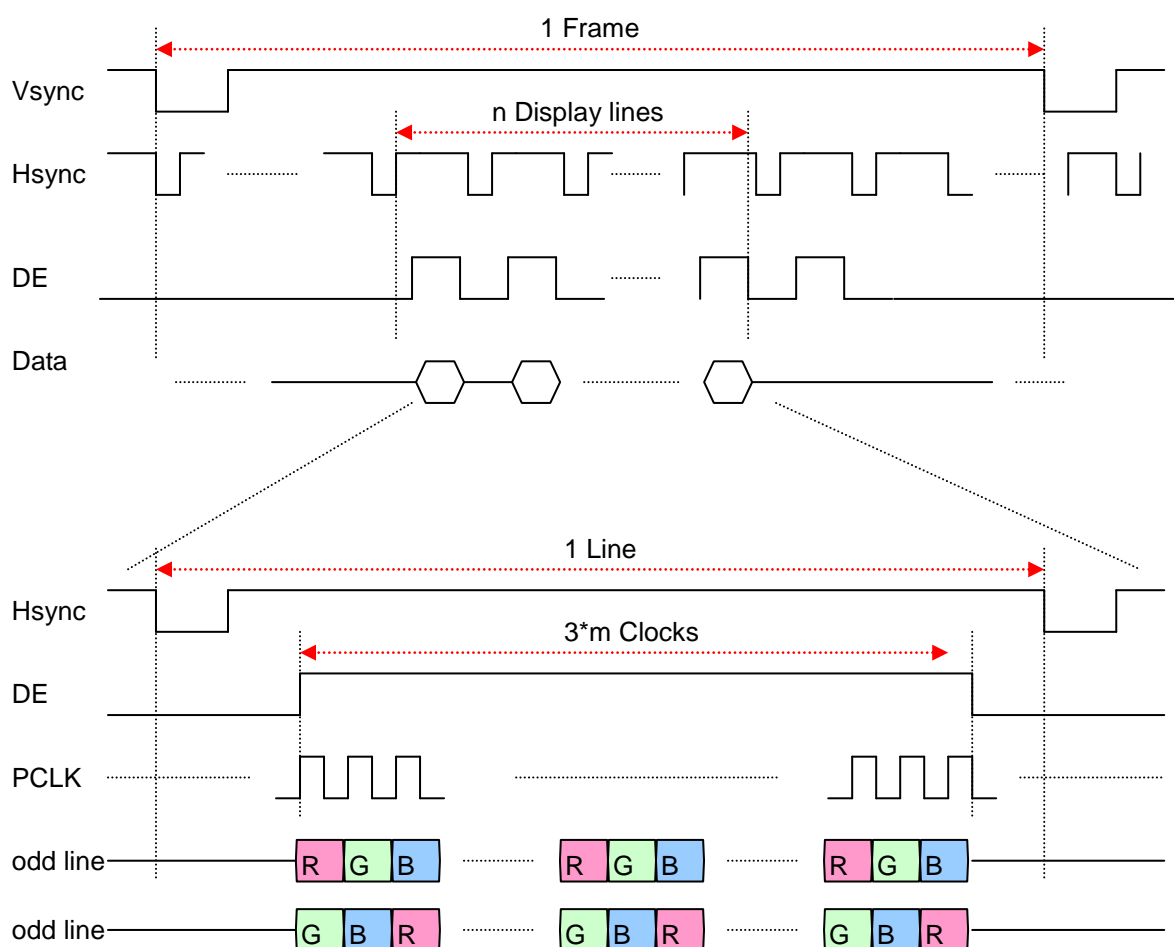
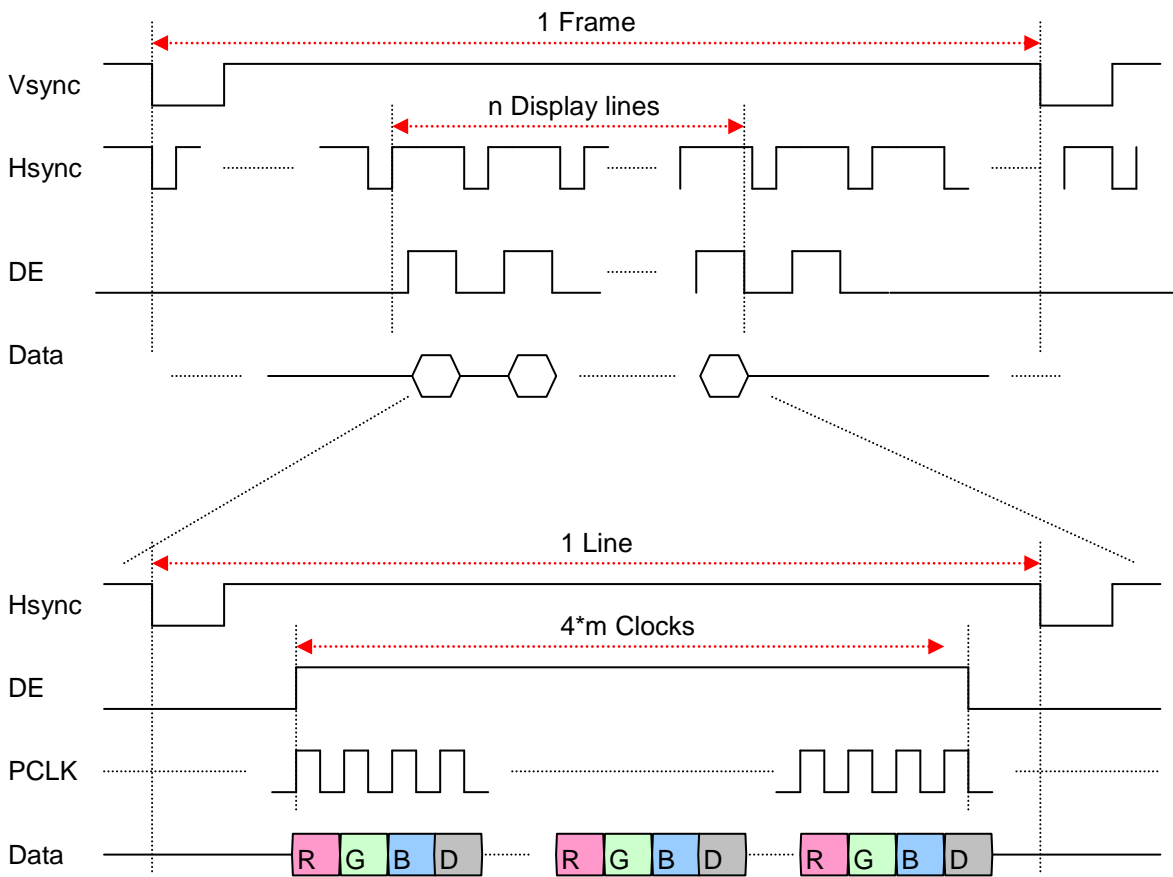


Figure 5-10 Delta RGB timing

5.8.5 RGB Dummy mode timing

This section shows the RGB Dummy diagram, the polarity of signal “Vsync”, “Hsync”, and “PCLK” can be programmed.



*Dummy = 0

Figure 5-11 RGB Dummy timing

5.9 Format of Frame Buffer

5.9.1 16bpp

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

5.9.2 18bpp

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	R5	R4	R3	R2	R1	R0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G5	G4	G3	G2	G1	G0	0	0	B5	B4	B3	B2	B1	B0	0	0

5.9.3 24bpp

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	R7	R6	R5	R4	R3	R2	R1	R0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

5.9.4 16bpp with alpha

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7	A6	A5	A4	A3	A2	A1	A0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R5	R4	R3	R2	R1	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1

5.9.5 18bpp with alpha

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7	A6	A5	A4	A3	A2	A1	A0	R5	R4	R3	R2	R1	R0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G5	G4	G3	G2	G1	G0	0	0	B5	B4	B3	B2	B1	B0	0	0

5.9.6 24bpp with alpha

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7	A6	A5	A4	A3	A2	A1	A0	R7	R6	R5	R4	R3	R2	R1	R0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

5.9.7 24bpp compressed

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLUE 1 [7:0]								RED 0 [7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GREEN 0 [7:0]								BLUE 0 [7:0]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GLEEN 2 [7:0]								BLUE 2 [7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED 1 [7:0]								GLEEN 1 [7:0]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RED 3 [7:0]								GLEEN 3 [7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLUE3 [7:0]								RED2 [7:0]							

5.10 Format of Data Pin Utilization

5.10.1 18-bit Parallel TFT

Col0 (RGB)																	
D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

5.10.2 16-bit Parallel TFT

Col0 (RGB)															
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

5.10.3 8-bit Serial TFT (24bpp)

Col0 (R)							
D7	D6	D5	D4	D3	D2	D1	D0
Col0 (G)							
D7	D6	D5	D4	D3	D2	D1	D0
Col0 (B)							
D7	D6	D5	D4	D3	D2	D1	D0

5.11 LCD Controller Operation

5.11.1 Set LCD Controller AHB Clock and Pixel Clock

The LCD Controller has 2 clock input: AHB clock and pixel clock. The both clocks are generated by CPM (Clock and Power Manager). The frequency of the 2 clocks can be set by CPM registers. `lcdc`'s AHB clock is equal to AHB0 clock (HCLK in CPM spec), and `CPM.LPCDR` set LCD pixel clock division ratio. Please refer to CPM spec for detail.

LCD AHB clock is the LCD controller's internal clock while LCD pixel clock is output to drive LCD panel. There have 2 rules for LCD clocks:

- 1 For TFT Panel, the frequency of LCD AHB clock must be at least 1.5 times of LCD pixel clock.
- 2 For STN Panel, the frequency of LCD AHB clock must be at least 3 times of LCD pixel clock.

LCD panel determines the frequency of LCD pixel clock.

5.11.2 Enabling the Controller

If the LCD controller is being enabled for the first time after system reset or sleep reset, all of the LCD registers must be programmed as follows:

- 1 Write the frame descriptors to memory.
- 2 Program the entire LCD configuration registers except the Frame Descriptor Address Registers (`LCDDAx`) and the LCD Controller enable bit (`LCDCTRL.ENA`).
- 3 Program `LCDDAx` with the memory address of the frame descriptor.
- 4 Enable the LCD controller by writing to `LCDCTRL.ENA`.

If the LCD controller is being re-enabled, there has not been a reset since the last programming; only the registers `LCDDAx` and `LCDCTRL.ENA` need to be reprogrammed. The LCD Controller Status Register (`LCDSTATE`) must also be written to clear any old status flags before re-enabling the LCD controller.

Once the LCD controller has been enabled, do not write new values to LCD registers except `LCDCTRL.ENA` or `DIS`.

5.11.3 Disabling the Controller

The LCD controller can be disabled in two ways: regular and quick.

- 1 Regular disabling.
Regular disabling is accomplished by setting the disable bit, `LCDCTRL.DIS`. The other bits in `LCDCTRL` must not be changed — read the register, set the `DIS` bit, and rewrite the register. This method causes the LCD controller to stop cleanly at the end of a frame. The LCD Disable Done bit, `LCDSTATE.LDD`, is set when the LCD controller finishes displaying the last frame, and the enable bit, `LCDCTRL.ENA`, is cleared automatically by hardware. `LCDCTRL.DIS` must be set zero when enabling the controller.

2 Quick disabling.

Quick disabling is accomplished by clearing the enable bit, LCDCTRL.ENA. The LCD controller will finish any current DMA transfer, stop driving the panel, setting the LCD Quick Disable bit (LCDSTATE.QD) and shut down immediately. This method is intended for situations such as a battery fault, where system bus traffic has to be minimized immediately so the processor can have enough time to store critical data to memory before the loss of power. The LCD controller must not be re-enabled until the QD bit is set, indicating that the quick shutdown is complete. Do not set the DIS bit when a quick disabling command has been issued.

NOTE: It is strongly recommended that software set the “LCD Module Stop Bit” in PMC to shut down LCDC clock supply to save power consumption after disable LCDC. Please refer to PMC for detailed information.

5.11.4 Resetting the Controller

At reset, the LCD Controller is disabled. All LCD Controller Registers are reset to the conditions shown in the register descriptions.

5.11.5 Frame Buffer

The starting address of frame buffer stored in external memory must be aligned to 4, 8 or 16 words boundary according to register LCDCTRL.BST. The length of buffer must be multiple of word (32-bit).

If LCDCTRL .BST = 0, align frame buffer to 16 word boundary

If LCDCTRL .BST = 1, align frame buffer to 8 word boundary

If LCDCTRL .BST = 2, align frame buffer to 4 word boundary

One frame buffer contains encoded pixel data of multiple of screen lines; each line of encoded pixel data must be aligned to word boundary. If the length of a line is not the multiple of word, extra bits must be applied to reach a word boundary. It is suggested that the extra bits to be set zero.

5.11.6 CCIR601/CCIR656

CCIR601: just as 16bit-parallel output.

CCIR656: need external encoder, or software designer need give digital blanking data and timing reference signal in data buffer.

5.11.7 OSD Operation

1 Normal process.

a Configuration.

* LCDRGB and LCDIPUR

- b Set Color.
 - * LCDBGC, LCDKEY0, LCDKEY1, LCDALHPA
- c Set Display.
 - * LCDVAT, LCDDAH, LCDDAV
 - * LCDVSYNC, LCDHSYNC
- d Set Descriptors.
 - * LCDIID
 - * LCDDA0, LCDSA0, LCDFID0, LCDCMD0, LCDOFFS0, LCDPW0, LCDCNUM0/LCDPOS0, LCDDSIZE0
 - * LCDDA1, LCDSA1, LCDFID1, LCDCMD1, LCDOFFS1, LCDPW1, LCDCNUM1/LCDPOS1, LCDDSIZE1
- e Enable LCDC.
- f Check the state from register LCDSTATE and LCDOSDS.

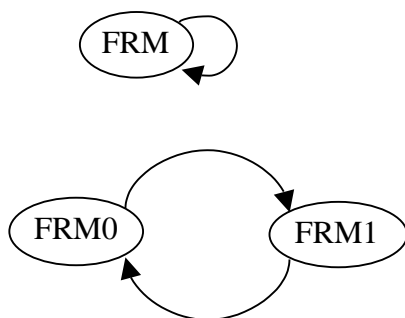
2 Reconfigure OSD.

If foreground0 and foreground1 (enable, position, size, bpp, alphamd, etc.) need to reconfigure during display process, just prepare a new descriptor and lcd will reconfigure automaticly.

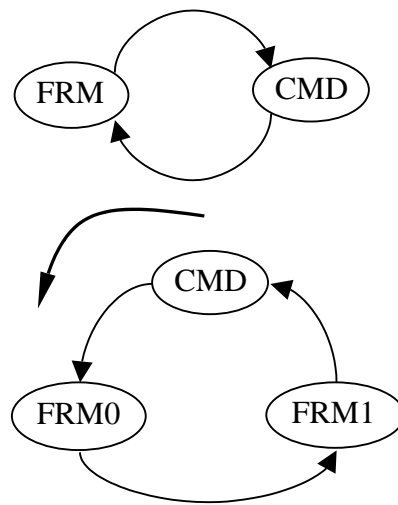
5.11.8 Descriptor Operation

1 TFT panel

you can use only one descriptor or several connected descriptor. As which shown below.



2 SLCD



5.11.9 IPU direct connect mode

When you use IPU direct connect mode, you just need to configure the foreground1 in IPU

* "IPU frame end flag" please refer to IPU spec.

5.11.10VGA output

When you use VGA output you need:

- 1 Open all channel of DAC. (refer to TVEDAC spec)
- 2 Set TVEN to 0.
- 3 Disable LCD panel pins (except HSYNC/VSYNC) for save power. (refer to GPIO spec)

6 Smart LCD Controller

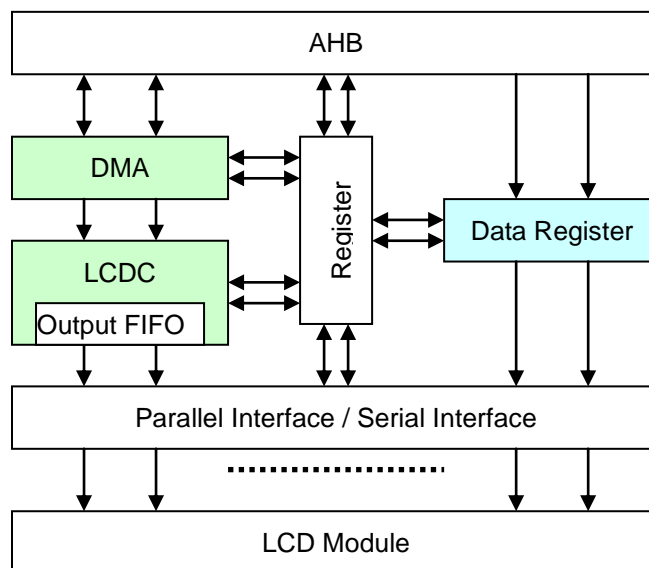
6.1 Overview

The Smart LCD Controller affords an interface to transfer data from the LCD controller to the LCD Module. It supports DMA operation and register operation.

Features:

- Supports a large variety of LCD Module from different vendors
- Supports parallel and serial interfaces
- Supports different size of display panel
- Supports different width of pixel data
- Supports internal DMA operation and register operation
- Supports Write Operation. Read Operation is not supported

6.2 Structure



*Please notice that the command only can transfer by DMA channel 0. No matter the DMA channel 1 or IPU are use or not.

6.3 Pin Description

Table 6-1 SLCD Pins Description

Name	I/O	Description	Interface
SLCD_DC	O	Command/Data Select Signal. The polarity of the signal can be programmable.	Serial: DC Parallel: DC
SLCD_WR	O	Data Sample Signal. The polarity of the signal can be programmable.	Serial: WR Parallel: Sample Data with the edge of WR
SLCD_CLK	O	The clock of SLCD. The polarity of the clock can be programmable.	Serial or not used
SLCD_DAT ^{*1} [17:0]	O	The data of SLCD. Relate to 1.9.4 Data mapping to GPIO function.	Serial: SLCD_DAT [15] Parallel: 18bit SLCD_DAT [17:0] 16bit SLCD_DAT [15:0] 8bit SLCD_DAT [7:0]
LCD_LO6_O	O	24 bit parallel SLCD RGB (or 24 bit command) low bit ([17:16],[9:8],[1:0]) output Relate to 1.9.4 Data mapping to GPIO function.	--

NOTE:

*1: SLCD_DAT [15] is also use as data pin for serial. The SLCD pins are shared with LCDC. You can see the set of register LCDCFG.LCDPIN in LCDC spec.

6.4 Register Description

In this section, we will describe the registers in Smart LCD controller. Following table lists all the registers definition. All register's 32bit address is physical address. And detailed function of each register will be described below.

Name	Description	RW	Reset Value	Address	Access Size
MCFG	SLCD Configure Register	RW	0x0000	0x130500A0	32
MCTRL	SLCD Control Register	RW	0x00	0x130500A4	8
MSTATE	SLCD Status Register	RW	0x00	0x130500A8	8
MDATA	SLCD Data Register	RW	0x00000000	0x130500AC	32

6.4.1 SLCD Configure Register (MCFG)

The register MCFG is used to configure SLCD.

MCFG															0x130500A0																
Bit															15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
															Reserved			DWIDTH			CWIDTH		Reserved			CSPLY	RSPLY	Reserved	CLKPLY	TYPE	
RST															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW																		
15:13	Reserved	Writing has no effect, read as zero.	R																		
12:10	DWIDTH ^{*1}	Data Width. <table border="1"> <thead> <tr> <th>DWIDTH</th> <th>Data Width</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>18-bit once Parallel/Serial</td> </tr> <tr> <td>001</td> <td>16-bit once Parallel/Serial</td> </tr> <tr> <td>010</td> <td>8-bit third time Parallel</td> </tr> <tr> <td>011</td> <td>8-bit twice Parallel</td> </tr> <tr> <td>100</td> <td>8-bit once Parallel/Serial</td> </tr> <tr> <td>101</td> <td>24-bit once Parallel</td> </tr> <tr> <td>111</td> <td>9-bit twice Parallel</td> </tr> <tr> <td>110</td> <td>Reserved</td> </tr> </tbody> </table> <p><i>*Please notice that you can only use 24-bit parallel command when use 24-bit parallel data. (The command may not 24-bit but need put them as 24-bit in memory(one command use one word))</i></p>	DWIDTH	Data Width	000	18-bit once Parallel/Serial	001	16-bit once Parallel/Serial	010	8-bit third time Parallel	011	8-bit twice Parallel	100	8-bit once Parallel/Serial	101	24-bit once Parallel	111	9-bit twice Parallel	110	Reserved	RW
DWIDTH	Data Width																				
000	18-bit once Parallel/Serial																				
001	16-bit once Parallel/Serial																				
010	8-bit third time Parallel																				
011	8-bit twice Parallel																				
100	8-bit once Parallel/Serial																				
101	24-bit once Parallel																				
111	9-bit twice Parallel																				
110	Reserved																				
9:8	CWIDTH ^{*1}	Command Width. <table border="1"> <thead> <tr> <th>CWIDTH</th> <th>Command Width</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>16-bit once / 9bit once</td> </tr> <tr> <td>01</td> <td>8-bit once</td> </tr> <tr> <td>10</td> <td>18-bit once</td> </tr> <tr> <td>11</td> <td>24-bit once</td> </tr> </tbody> </table> <p><i>*Please notice that you can only use 24-bit parallel command when use 24-bit parallel data. (The command may not 24-bit but need put them as 24-bit in memory (one command use one word))</i></p>	CWIDTH	Command Width	00	16-bit once / 9bit once	01	8-bit once	10	18-bit once	11	24-bit once	RW								
CWIDTH	Command Width																				
00	16-bit once / 9bit once																				
01	8-bit once																				
10	18-bit once																				
11	24-bit once																				
7:5	Reserved	Writing has no effect, read as zero.	R																		
4	CSPLY	WR Polarity. (WR initial level will be different from CS Polarity) 0: Active Level is Low 1: Active Level is High	RW																		
3	RSPLY	DC Polarity. 0: Command DC = 0, Data DC = 1 1: Command DC = 1, Data DC = 0	RW																		

2	Reserved	Writing has no effect, read as zero.	R
1	CLKPLY	LCD_CLK Polarity. 0: Active edge is Falling 1: Active edge is Rising	RW
0	TTYPE	Transfer Type. 0: Parallel 1: Serial	RW

NOTE:

*1: The set of DWIDTH and CWIDTH should keep to the rules as follows:

Interface Mode	Command Width	Data Width	Color
Parallel	18-bit	18-bit once	R6G6B6
	16-bit	16-bit once	R5G6B5
		9-bit twice	--
	9-bit	9-bit twice	--
	8-bit	8-bit once	--
		8-bit twice	--
8-bit third times		--	
Serial	18-bit	18-bit once	--
	16-bit	16-bit once	--
	9-bit	9bit twice	--
	8-bit	8-bit once	--
		8-bit twice	--
		8-bit third times	--

6.4.2 SLCD Control Register (MCTRL)

MCTRL is SLCD Control Register.

MCTRL																0x130500A4							
Bit																7	6	5	4	3	2	1	0
																Reserved				DMASTART	DMAMODE	Reserved	
RST																0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:3	Reserved	Writing has no effect, read as zero.	R
2	DMAMODE	SLCD descriptor DMA mode select. 0: DMA will continually transfer data follow descriptor chain	RW

		1: DMA will stop when one descriptor finished	
1	DMASTART	Only use when DMAMODE = 1, set 1 to restart DMA transfer.	RW
0	Reserved	Writing has no effect, read as zero.	R

6.4.3 SLCD Status Register (MSTATE)

The register of MSTATE is SLCD status register.

MSTATE																0x130500A8							
Bit																7	6	5	4	3	2	1	0
	Reserved														Reserved				BUSY				
RST																0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:1	Reserved	Writing has no effect, read as zero.	R
0	BUSY	Transfer is working or not. This bit will be set to 1 when transfer is working. It will be cleared by hardware when transfer is finished. 0: not busy 1: busy	RW

6.4.4 SLCD Data Register (MDATA)

The register MDATA is used to send command or data to LCM. When RS=0, the low 24-bit is used as command. When RS=1, the low 24-bit is used as data.

MDATA																0x130500AC																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	RS	Reserved								DATA / CMD																							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW
31	RS	The RS bit of data register is used to decide the meanings of the low 24-bit. 0: data 1: command	RW

30:24	Reserved	Writing has no effect, read as zero.	R
23:0	DATA/CMD	Data or Command Register.	RW

6.5 System Memory Format

6.5.1 Data format

You can configure these registers according to LCDC module.

6.5.2 Command Format

1 18-bit command

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
X	X	X	X	X	X	X	X	X	X	X	X	X	X	C17	C16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

2 16-bit command

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

3 9-bit command once

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
X	X	X	X	X	X	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	X	X	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

4 8-bit command once

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C7	C6	C5	C4	C3	C2	C1	C0	C7	C6	C5	C4	C3	C2	C1	C0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C7	C6	C5	C4	C3	C2	C1	C0	C7	C6	C5	C4	C3	C2	C1	C0

5 8-bit command twice (Command = command part + data part)

*Please notice that when you use this kind command, set CWIDTH as 8bit once and set the LCDCNUM.CNUM as doubled the real command number.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D7	D6	D5	D4	D3	D2	D1	D0	C7	C6	C5	C4	C3	C2	C1	C0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0	C7	C6	C5	C4	C3	C2	C1	C0

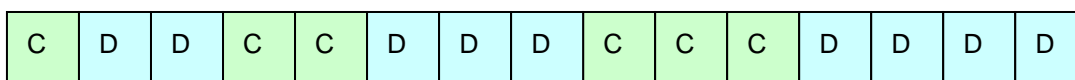
6.6 Transfer Mode

Two transfer modes can be used: DMA/IPU Transfer Mode and Data Register Transfer Mode. In DMA/IPU mode, always transfer commands by DMA 0.

6.6.1 DMA Transfer Mode

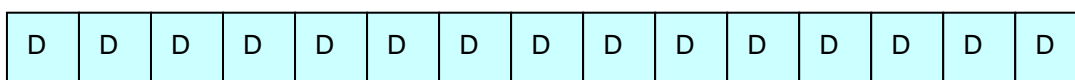
Command and data can be recognized by RS bit coming from memory. The format of DMA transfer can be as follows:

1 Command and Data



*Please notice that the command only can insert between two complete frame and the number of command is 0~255.

2 Only Data



*You can also not use command but you still need to use a command descriptor and set the CNUM = 0.

Because DMA transfer mode only can work in OSD mode, you need to configure the panel according OSD mode:

1 Configuration.

* LCDRGBC and LCDIPUR

2 Set Color.

* LCDBGC, LCDKEY0, LCDKEY1, LCDALHPA

3 Set Display.

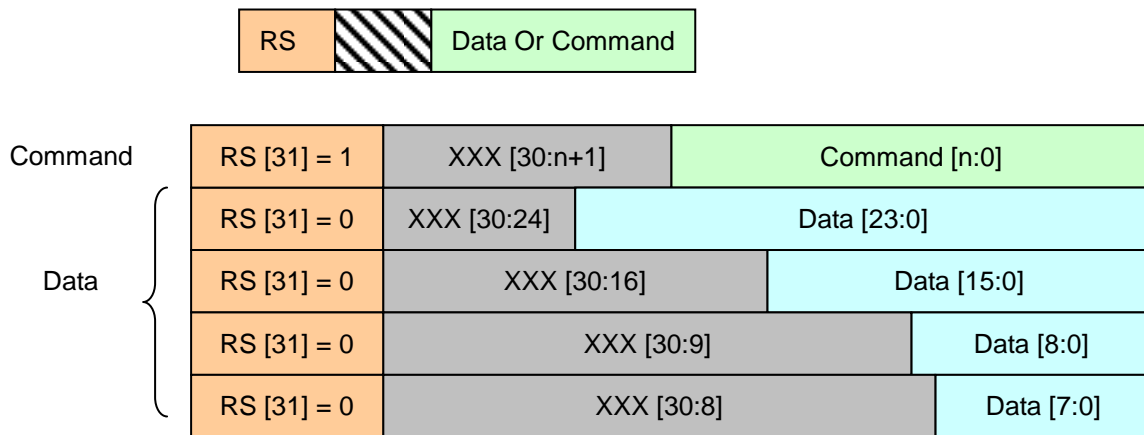
* LCDVAT, LCDDAH, LCDDAV

* LCDVSYNC, LCDHSYNC

- 4 Set Descriptors.
 - * LCDIID
 - * LCDDA0, LCDSA0, LCDFID0, LCDCMD0, LCDOFFS0, LCDPW0, LCDCNUM0/LCDPOS0, LCDDSIZE0
 - * LCDDA1, LCDSA1, LCDFID1, LCDCMD1, LCDOFFS1, LCDPW1, LCDCNUM1/LCDPOS1, LCDDSIZE1
- 5 Enable slcd DMA.
- 6 Enable LCDC.

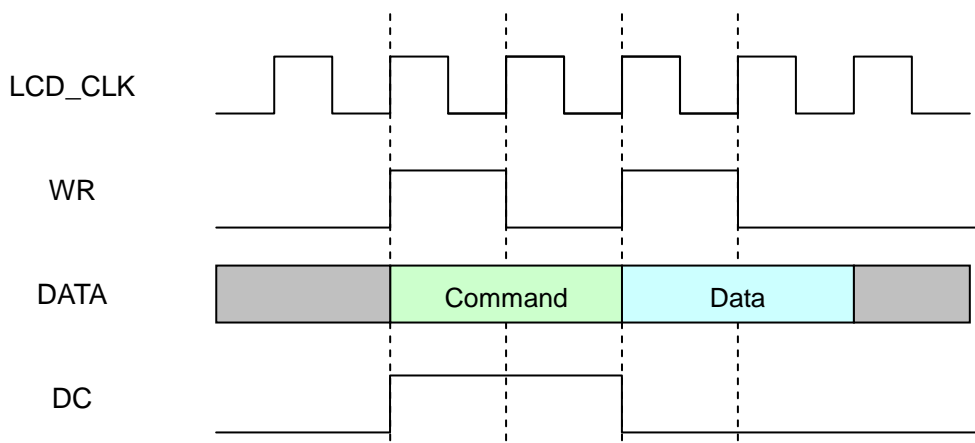
6.6.2 Register Transfer Mode

Each time you can write a command or a data to the register, then it will transfer the RS signal and data or command to LCM. Command and data can be recognized by RS bit coming from data register. The format of data register transfer can be as follows:

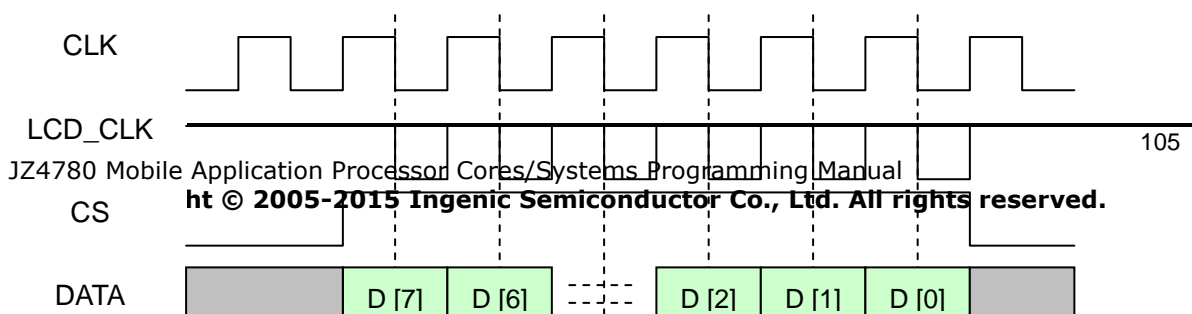


6.7 Timing

6.7.1 Parallel Timing



6.7.2 Serial Timing



6.8 Operation Guide

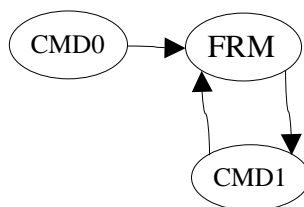
6.8.1 DMA Operation

1 Start DMA transfer.

- a Set LCDCFG.MODE to 1101 to choose LCM.
- b Set LCDCTRL.BST to choose burst length for transferring.
- c Set register LCDIID0, LCDDA0, LCDSA0, LCDFID0, LCDCMD0, LCDOFFS0, LCDPW0, LCDCNUM0, LCDESSIZE0 to initial internal DMA.
- d Also set register LCDIID1, LCDDA1, LCDSA1, LCDFID1, LCDCMD1, LCDOFFS1, LCDPW1, LCDCNUM1, LCDESSIZE1 when use DMA channel 1 in OSD mode.
- e Set MCFG to configure SLCDC.
- f Before starting DMA, Wait for MSTATE.BUSY == 0.
- g Set MCTRL.DMATXEN to 1 to prepare DMA transfer.
Note that if you don't want to stop DMA transfer, you need not to check MSTATE.BUSY.
- h Set LCDCCTRL.ENA to 1 to start LCDC internal DMA.
- i The LCDC internal DMA will transfer data to SLCDC, and SLCDC transfer data to LCM.
Repeat this step till you want to close the SLCDC to transfer data to LCM Panel.

*please notice that use and only use DMA0 to transfer command no matter use DMA0 to transfer frame data or not.

One recommend descriptor chain (CMD0 with CNUM>0 and CMD1 with CNUM=0):



2 Stop DMA transfer.

- a Set LCDCCTRL.ENA to 0 to stop LCDC internal DMA at once.
- b Wait till MSTATE.BUSY is set to 0 by hardware.

- MSTATE.BUSY == 1: there is data in the FIFO waited for transferring to LCM.
MSTATE.BUSY == 0: all data in the FIFO have finished transferring to LCM.
- c Set MCTRL.DMATXEN to 0 to stop DMA transfer.

3 Restart DMA transfer.

When MCTRL.DMATXEN is set to 0, and then you want to restart DMA transfer at once, you should ensure that MCTRL.DMATXEN must keep 0 at least three cycles of PIXCLK.

6.8.2 Register Operation

- 1 Set MCFG to configure SLCD.
- 2 Wait for MSTATE.BUSY == 0.
- 3 Set MDATA register.
- 4 Wait for MSTATE.BUSY == 0.
- 5 Set MDATA register.
- 6 Wait for MSTATE.BUSY == 0.
- 7

7 Decompressor

7.1 Overview

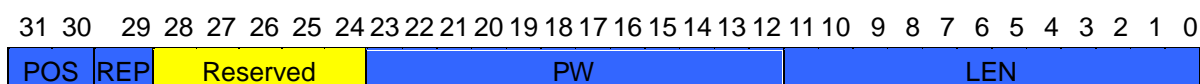
- Support bpp16 compressed data
- Support bpp24 compressed data with alpha
- Support bpp24 compressed data without alpha

7.2 Compress Method

Components

There are two kinds of Components in this method , one is command, the other is data .

Command:



Bits	Name	Description										
31:30	POS	Position flag, indicates this command is in the head or middle or end of a giving line.										
		<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>2'b00</td> <td>Reserved</td> </tr> <tr> <td>2'b01</td> <td>This is the last command of the line</td> </tr> <tr> <td>2'b10</td> <td>This is sequent command of the line</td> </tr> <tr> <td>2'b11</td> <td>The is the first command of the line</td> </tr> </tbody> </table>	Value	Description	2'b00	Reserved	2'b01	This is the last command of the line	2'b10	This is sequent command of the line	2'b11	The is the first command of the line
		Value	Description									
		2'b00	Reserved									
		2'b01	This is the last command of the line									
2'b10	This is sequent command of the line											
2'b11	The is the first command of the line											
29	REP	Repeated flag, indicates the following data is/are repeated or not , in another word , indicates the following data is/are compressed or original. 0: repeated 1: non-repeated										
28:24	Reserve	All zero.										
23:12	PW	Page width of current line. In the start cmd, it Indicates how many words in this line. In other cmd, it is zero.										
11:0	LEN	Length , in repeated case, it indicates how many times the following data repeated; in non-repeated case , it indicates the number of following data frames. <i>*in the case of compressed without alpha bpp24, this should be all zero.</i>										

**In the case of compressed without alpha bpp24, there is just start command , no sequent command and last command, the length is the high 8bits of every data.*

Data:

In the case of compressed without alpha bpp24, the length is the high 8bits of every data , other bits are original data.

In the case of compressed with alpha bpp24, the data is ARGB8888.

In the case of compressed bpp16, the data is RGB565 or RGB555, and should be word aligned.

Example

There is an example, in next page.

The first one of the following figures is compressed with alpha;

The second one of the following figures is compressed without alpha.

That is 7 x 4 picture.

The dummies will not be fetched by DMA.

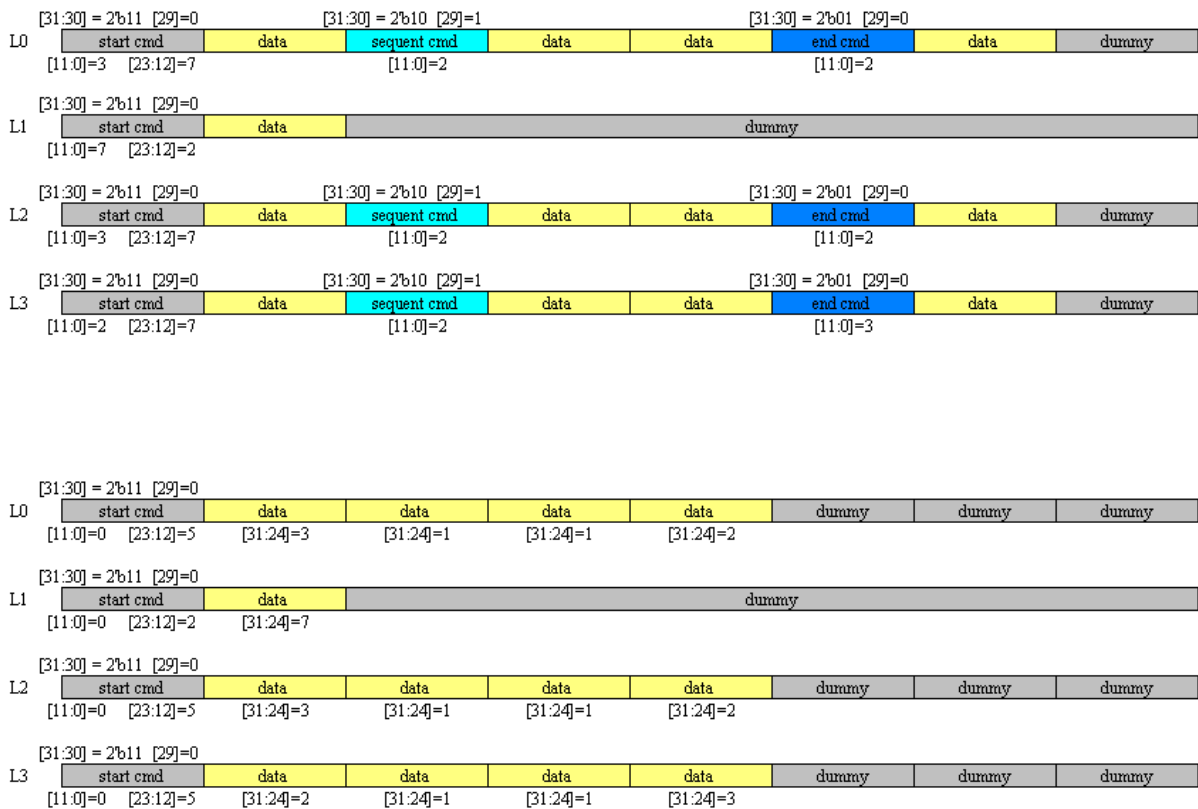
The first line contains 3 repeated pixels, 2 non-repeated pixels, and 2 repeated pixels;

The second contains 7 repeated pixels;

The third line contains 3 repeated pixels , 2 non-repeated pixels, and 2 repeated pixels;

The last line contains 2 repeated pixels , 2 non-repeated pixels, and 3 repeated pixels.

:



7.3 Operation Guide

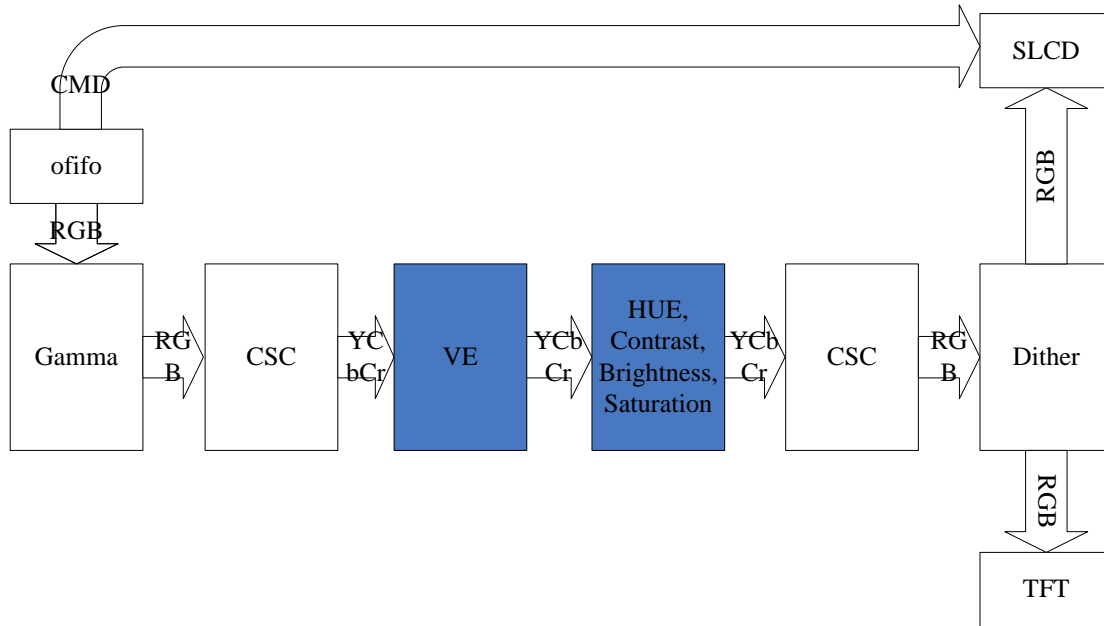
The routine operations are the same as normal operations except descriptor operation:

- 1 Uncomp_en (LCDCMDx[27]) should be set if decompressing function is used.
- 2 Uncomp_md(LCDCMDx[26]) should be set if the frame is bpp24 without alpha compressed frame, otherwise, should be clear.
- 3 LCDPWx don't need to set value.
- 4 LEN (LCDCMDx[11:0]) should be set to the LPP value of the frame.
- 5 LCDOFFSx should be set to how many word in per line of frame buffer for compressed frame, count in word, 64-word align or 16-word align(depend on the configuration of aosd_comp).

8 Image Enhancement

8.1 Overview

Image Enhancement Includes Gamma , CSC, VEE, Contrast, Brightness, HUE, Saturation and Dither. And the blue area in the following figure works in YCbCr color space.



8.2 Operation Guide

1. Read corresponding bit in LCDENH_STATUS. If it's 1 then continue, otherwise disable corresponding function and wait it change to 1.
2. Configure corresponding registers.
3. Enable corresponding function.

9 Image Process Unit

9.1 Overview

IPU (Image process unit) contains Resize and CSC (color space conversion), which is used for image post processing.

9.1.1 Feature

- Location: AHB bus
- Input format
 - Separate frame: YUV /YCbCr (4:2:0, 4:2:2, 4:4:4, 4:1:1), RGB888
 - Packaged data: YUV422, RGB888, RGB565, RGB555, YUV444
 - Separate frame in block format: YUV/YCbCr 420
- Output data format
 - RGB (565, 555, 888, AAA)
 - Packaged data YUV422
- Color convention coefficient: configurable (CSC enable)
- Minimum input image size (pixel): 4x4
- Maximum input image size (pixel): 8096x8096
- Maximum output image size (pixel)
 - Width: up to 4095 (without vertical resizing)
up to **2048** (with vertical resizing)
 - Height: up to 4095
- Image resizing
 - Support bilinear
 - 0 and bi-cube zooming mode
 - Up scaling ratios up to 1:31 in fractional steps with 1/32 accuracy
 - Down scaling ratios up to 31:1 in fractional steps with 1/32 accuracy

**For more details, refer to Special Instruction.*

9.2 Block

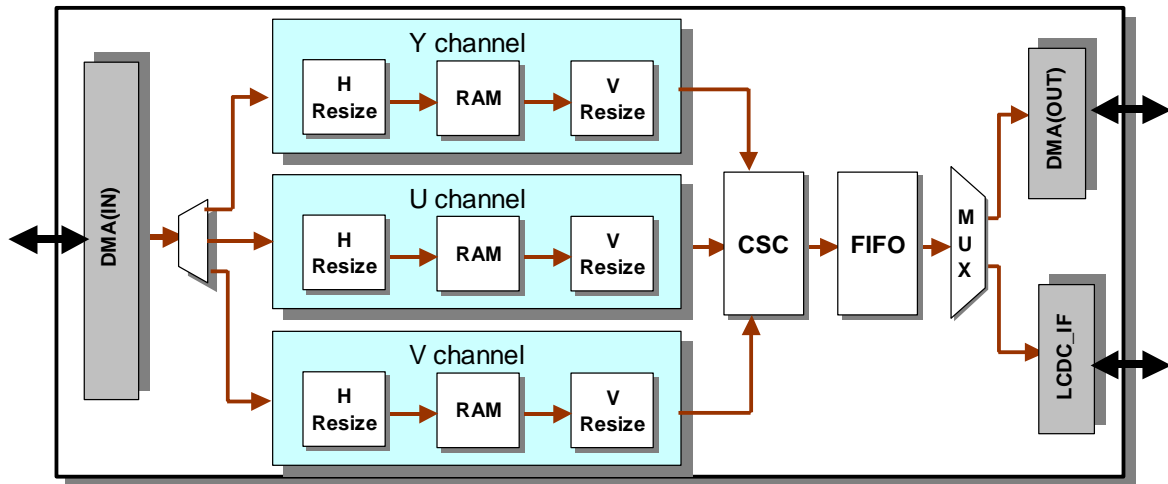


Figure 9-1 The Block about the IPUData flow

9.3 Data flow

9.3.1 Input data

- Separated YUV (or YcbCr/RGB888; the following use YUV for convenience) Frame case: Y, U, V data would be fetched from external memory by DMA burst read operation.
- Packaged YUV422 (or RGB888/ RGB565/ RGB555/YUV444) case: Packaged YUV(RGB888/ RGB565/ RGB555/YUV444) data would be fetched from external memory by DMA burst read operation.

9.3.2 Output data

- DMA output to external memory case: The output data format could be RGB (565, 555, 888) or YUV (package422), and the data would be stored to the external memory by DMA burst write operation.
- Flow into LCDC case: The output data format can be RGB or YUV (package), and the transfer would not cross AHB BUS.

9.3.3 Resize Coefficients LUT

The resize coefficients look up table is preset by software according to specific format (see for 9.4.32, 9.4.33 detail). There are 2 tables to support independent horizontal and vertical scaling. Each table has 32 entries that can accommodate up to 32 coefficient-groups.

9.4 Registers Descriptions

The physical address base for the address-mapped registers of IPU is **0x13080000**.

The following table will show all the register address.

Table 9-1 register list

NAME	Offset	Descript
<u>IPU_FM_CONTROL</u>	0x0	IPU frame controller
<u>IPU_STATUS</u>	0x4	IPU global status register
<u>D_FMT</u>	0x8	IPU data format register
<u>Y_ADDR</u>	0xC	Source Y (or R) base address
<u>U_ADDR</u>	0x10	Source U (or G) base address
<u>V_ADDR</u>	0x14	Source V (or B) base address
<u>IN_FM_GS</u>	0x18	Input Geometric Size (height and width)
<u>Y_STRIDE</u>	0x1C	Source Y frame stride
<u>UV_STRIDE</u>	0x20	Source U and V frame stride
<u>OUT_ADDR</u>	0x24	Result frame base address
<u>OUT_GS</u>	0x28	Result frame size (height and width)
<u>OUT_STRIDE</u>	0x2C	Result frame stride
<u>RSZ_COEF_INDEX</u>	0x30	Resize Coefficients Table Index
<u>CSC_C0_COEF</u>	0x34	Color conversion Coefficient
<u>CSC_C1_COEF</u>	0x38	Color conversion Coefficient
<u>CSC_C2_COEF</u>	0x3C	Color conversion Coefficient
<u>CSC_C3_COEF</u>	0x40	Color conversion Coefficient
<u>CSC_C4_COEF</u>	0x44	Color conversion Coefficient
<u>HRSZ_COEF_LUT</u>	0x48	Horizontal Resize Coefficients Look Up Table
<u>VRSZ_COEF_LUT</u>	0x4C	Vertical Resize Coefficients Look Up Table
<u>CSC_OFFSET_PARA</u>	0x50	Color conversion offset Coefficient
<u>SRC_TLB_ADDR</u>	0x54	Base address of the source Y's physical address map table
<u>DEST_TLB_ADDR</u>	0x58	Base address of the destination's physical address map table
<u>IPU_ADDR_CTRL</u>	0x64	IPU address set controller
<u>Y_ADDR_N</u>	0x84	Source Y base address of next frame
<u>U_ADDR_N</u>	0x88	Source U base address of next frame
<u>V_ADDR_N</u>	0x8C	Source V base address of next frame
<u>OUT_ADDR_N</u>	0x90	Result frame base address of next frame
<u>SRC_TLB_ADDR_N</u>	0x94	Base address of the source Y's physical address map table for next frame
<u>DEST_TLB_ADDR_N</u>	0x98	Base address of the destination's physical address map table for next frame
<u>PIC_ENC_TABLE</u>	0x400 ~0X7FF	Picture enhance table stone.
<u>REG_EN_MSK</u>	0x70	Register enable indicator
<u>TRIG</u>	0x74	lpu trigger register

FM_XYOFT	0x78	Frame display x and y offset in lcd screen
GLB_CTRL	0x7C	Ipu global control
OSD_CTRL	0x9C	Osd controller about IPU

9.4.1 IPU global Control Register

GLB_CTRL																0x7C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																DMA_OPT_EN		DMA_PRI		IRQ_EN											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
3	DMA_OPT_EN	Ipu auto priority option enable	R
2:1	DMA_PRI	IPU's priority on system bus 0: lowest 1: 2: 3: highest	RW
0	IRQ_EN	Interrupt enable 0: disable 1: enable	R/W

9.4.2 IPU OSD Control Register

OSD_CTRL																0x9C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																GALPHA						Reserved				OSD_PM		OSD_MOD			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	Reserved	Writing has no effect, read as zero.	R
15:8	GALPHA	Global alpha	RW
7:2	Reserved	Writing has no effect, read as zero.	R
2	OSD_PM	IPU layer has been pre-multiply 0: without 1: has pre-multiply	RW
1:0	OSD_MOD	Osd mode selector:	R/W

9.4.3 IPU Frame Control Register

IPU_FM_CONTROL																0x0																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	Reserved													ZOOM_SEL	DFIX_SEL	FIELD_SEL	FIELD_CONF	DISP_SEL	DPAGE_MAP	SPAGE_MAP	LCDC_SEL	SPKG_SEL	Reserved						FM_IRQ_EN	CSC_EN	VRSZ_EN	HRSZ_EN	Reserved	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		

Bits	Name	Description	R/W
31:19	Reserved	Writing has no effect, read as zero.	R
18	ZOOM_SEL	IPU rooming mode selector. 1: bi-cube 0: bilinear	RW
17	DFIX_SEL	Fixed destination address choose. (Valid when LCDC_SEL == 0) 0: not use the fixed address 1: use the fixed address	RW
16	FIELD_SEL *1	Destination field choose. (Valid when FIELD_CONF_EN == 1) 0: top field 1: bottom field	RW
15	FIELD_CONF_EN *1	FIELD_SEL mask. 0: do not change FIELD_SEL 1: configure FIELD_SEL Read as zero.	W
14	DISP_SEL	Destination display choose. 0: frame display mode 1: field display mode	RW
13	DPAGE_MAP	Destination address page mapping choose. 0: not use the page mapping 1: use the page mapping	RW
12	SPAGE_MAP	Source address page mapping choose. 0: not use the page mapping 1: use the page mapping	RW
11	LCDC_SEL	Output data destination choose. 0: output to external memory 1: output to LCDC FIFO	RW
10	SPKG_SEL	Input data case choose. 0: Separated Frame 1: Packaged Frame	RW

9:6	Reserved	Writing has no effect, read as zero.	R
5	FM_IRQ_EN	Frame process finish interrupt enable. 1: enable; 0: disable.	RW
4	CSC_EN	CSC enable. 1: enable; 0: disable.	RW
3	VRSZ_EN	Vertical Resize enable. 1: enable; 0: disable.	RW
2	HRSZ_EN	Horizontal Resize enable. 1: enable; 0: disable.	RW
1: 0	Reserved	Writing has no effect, read as 1.	R

NOTES:

- *1: The FIELD_SEL will work when the DISP_SEL is 1, which indicates the IPU is under the field display mode. And the IPU will output the picture from the initial field (top or bottom) to the next field (bottom or top) automatically when the current field has been outputted or stopped. The initial field can be configured by setting the FIELD_SEL to 0 or 1 with FIELD_CONF_EN is 1. The FIELD_CONF_EN is just the trigger that controls the FIELD_SEL's valuation.

9.4.4 IPU Status Register

IPU_STATUS																0x4																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																MSTATUS		Reserved	SIZE_ERR	Reserved	OUT_END										
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

Bits	Name	Description	R/W
31:6	Reserved	Writing has no effect, read as zero.	R
5:4	MSTATUS	IPU main status. 00: IPU is free others: IPU is running now.	R
3	Reserved	Writing has no effect, read as zero.	R
2	SIZE_ERR	The size error flag. 1: size error; 0: size ok.	R
1	Reserved	Writing has no effect, read as zero.	R
0	OUT_END *1	Output termination flag. 1: finished; 0: not finished.	R/W

NOTE:

- *1: If IPU_CONTROL.FM_IRQ_EN has been set 1, once OUT_END is set value 1 which denotes a frame's post process done, an low level active interrupt request will be issued until corresponding software handler read IPU_STATUS and clean end flag.

When the IPU_CONTROL.FM_LCDC_SEL has been set 1, and the IPU has finished one transfer, the LCDC and CPU need to occupy the IPU control. The IPU will monitor the request

signal from LCDC and the read signal from the CPU, then it will determine whether re-configure itself by the CPU if the CPU read first or output the same frame to LCDC again if the LCDC get the control.

9.4.5 IPU trigger register

IPU_TRIGGER																0x74																																															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
	reserved																																																														
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1																												

Bits	Name	Description	R/W
31:6	Reserved	Writing has no effect, read as zero.	R
3	RESET	reset	RW
2	STOP_LCD	Stop ipu to lcd, ipu will not output to lcd next frame	RW
1	STOP	Stop ipu, ipu will stop immediately in safe	W
0	RUN	New fame configurable need run.	W

9.4.6 IPU register configure enable control register

IPU_REG_CTRL																0x64																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved													OSD_RY	OFT_RY	VRC_RY	HRC_RY	RCT_RY	CSCC_RY	DSTR_RY	DGS_RY	CSTR_RY	YSTR_RY	IGS_RY	DF_RY	CTRL_RY	PTD_RY	PTS_RY	D_RY	V_RY	U_RY	Y_RY	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Bits	Name	Description	R/W
31:6	Reserved	Writing has no effect, read as zero.	R
18	OSD_RY	New OSD_CTRL ready.	RW
17	OFT_RY	new FM_XYOFT ready.	RW
16	VRC_RY	Vertical resize coefficient ready.	RW
15	HRC_RY	Horizontal resize coefficient ready.	RW
14	RCT_RY	RSZ_COEF_INDEX ready.	RW
13	CSCC_RY	CSC_C0(1,2,3)_COEF and CSC_OFSET_PARA ready.	RW
12	DSTR_RY	New destination stride (OUT_STRIDE) ready.	RW
11	DGS_RY	New destination geometric size (OUT_GS) ready.	RW

10	CSTR_RY	New UV stride (UV_STRIDE) ready	RW
9	YSTR_RY	New Y stride (Y_STRIDE) ready	RW
8	IGS_RY	New input geometric size (IN_FM_GS) ready.	RW
7	DF_RY	New data format (D_FMT) ready.	RW
6	CTRL_RY	New frame control (IPU_FM_CONTROL) ready	RW
5	PTD_RY	New destination TLB base address ready.	RW
4	PTS_RY	New source TLB base address ready.	RW
3	D_RY	New destination address ready.	RW
2	V_RY	New source V address ready.	RW
1	U_RY	New source U address ready.	RW
0	Y_RY	New source Y address ready.	RW

NOTES:

- When the xx_RY bit has been set 1, the IPU will use the new corresponding register in the next frame, and use the old address value whose corresponding bit in **IPU_REG_CTRL** is 0.

9.4.7 Data Format Register

D_FMT																0x8																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved							RGB_888	RGB_OUT_OFT			OUT_FMT			YUV_PKG_OUT_OFT			Reserved							BLK_SEL	IN_OFT		IN_FMT				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:26	Reserved	Writing has no effect, read as zero.	R
25	RGB_888_OUT_FMT	RGB888 output format indicator. (only used in RGB888 out OUT_FMT == 010) 0: the low 24 bits will be the pixel in a word 1: the high 24 bits will be the pixel in a word	RW
24:22	RGB_OUT_OFT	Output data packaged offset. (only used in RGB out OUT_FMT != 011) [31:0] 000: RGB 001: RBG 010: GBR 011: GRB 100: BRG 101: BGR Others: reserved	RW
21:19	OUT_FMT	Indicates the destination data format.	RW

		000: RGB555 001: RGB565 010: RGB888 011: YUV422 package 100: RGBAAA(R(or G or B) is 10 bits wide)	
18:16	YUV_PKG_OUT_OFT	Output data packaged offset. (only used in CSC disable case and in the source is YUV422 packaged case (IPU_CONTROL.SPKG_SEL == 1)) [31:0] 000: Y1UY0V 001: Y1VY0U. 010: UY1VY0 011: VY1UY0 100: Y0UY1V 101: Y0VY1U 110: UY0VY1 111: VY0UY1	RW
15: 6	Reserved	Writing has no effect, read as zero.	R
4	BLK_SEL	Indicate the source data format when source is YUV420	RW
3:2	IN_OFT	Input data packaged offset. (when the source is YUV422 packaged case (IPU_CONTROL.SPKG_SEL == 1 && IN_FMT == 01)) 00: Y1UY0V 01: Y1VY0U 10: UY1VY0 11: VY1UY0	RW
1:0	IN_FMT	Indicates the source data format. When IPU_CONTROL.SPKG_SEL == 0 00: YUV 4:2:0 01: YUV 4:2:2 10: YUV 4:4:4 11: YUV 4:1:1 When IPU_CONTROL.SPKG_SEL == 1 00: RGB555 01: YUV 4:2:2 10: RGB888 or YUV444 11: RGB565	RW

NOTES:

- 1 When the source frame is packed RGB, the **IPU_CONTROL.SPKG_SEL** must be 1 and **IN_FMT** must be 10, and when the source frame is packed YUV, the **IPU_CONTROL.SPKG_SEL** must be 1 and **IN_FMT** must be 01.
- 2 When the source frame is packed YUV444, the data format about a pixel should be as following:

31	24	23	16	15	8	7	0
0	Y	U	V				

9.4.8 Input Y Data Address Register

	Y_ADDR	0xc
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	

	Y_ADDR																												0	0				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	Y_ADDR *1	In separated Frame case, it indicates the source Y (or R) data buffer's start address. In source YUV422 package case, it indicates the start Address of the packaged Frame.	RW

NOTES:

- 1 When the `IPU_CONTROL.SPAGE_MAP == 1`, the Y_ADDR should be the start virtual address.
- 2 Y_ADDR should be **word align**.

9.4.9 Input U Data Address Register

	U_ADDR														0x10																			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	U_ADDR																												0	0				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	U_ADDR *1	The source U data buffer's start address of separated frame case.	RW

NOTES:

- 1 When the `IPU_CONTROL.SPAGE_MAP == 1`, the U_ADDR should be the start virtual address.
- 2 U_ADDR should be **word align**.

9.4.10 Input V Data Address Register

	V_ADDR														0x14																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

V_ADDR																												0	0		
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	V_ADDR	The source V data buffer's start address of separated Frame case.	RW

NOTES:

- 1 When the IPU_CONTROL.SPAGE_MAP == 1, the V_ADDR should be the start virtual address.
- 2 V_ADDR should be word align.

9.4.11 Input source TLB base address

SRC_TLB_ADDR																0x54																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SRC_TLB_ADDR																												0	0				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	SRC_TLB_ADDR	The SOURCE TLB base address. (This register will act when the IPU_CONTROL.PAGE_MAP==1)	RW

9.4.12 Destination TLB base address

DEST_TLB_ADDR																0x58																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DEST_TLB_ADDR																												0	0				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	DEST_TLB_ADDR	The destination frame's TLB base address. (This register will work when the IPU_CONTROL.DPAGE_MAP==1)	RW

9.4.13 Input Y Data Address of next frame Register

Y_ADDR_N																0x84																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Y_ADDR_N																														0	0	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	Y_ADDR_N	In separated Frame case, it indicates the source Y (or R) data buffer's start address of the next frame. In package case, it indicates the start address of the packaged Frame of the next frame.	R

9.4.14 Input U Data Address of next frame Register

U_ADDR_N																0x88																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	U_ADDR_N																														0	0	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	U_ADDR_N	The source U (G) data buffer's start address of the next frame in separated frame case.	R/W

NOTE: When the IPU_CONTROL.SPAGE_MAP == 1, the U_ADDR_N will be the start virtual address.

9.4.15 Input V Data Address of next frame Register

V_ADDR_N																0x8C																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	V_ADDR_N																														0	0	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	V_ADDR_N	The source V (B) data buffer's start address of the next frame in separated frame case.	RW

NOTE: When the IPU_CONTROL.SPAGE_MAP == 1, the V_ADDR_N will be the start virtual address.

9.4.16 Source TLB base address of next frame

SRC_TLB_ADDR_N																0x94																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRC_TLB_ADDR_N																0	0														
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

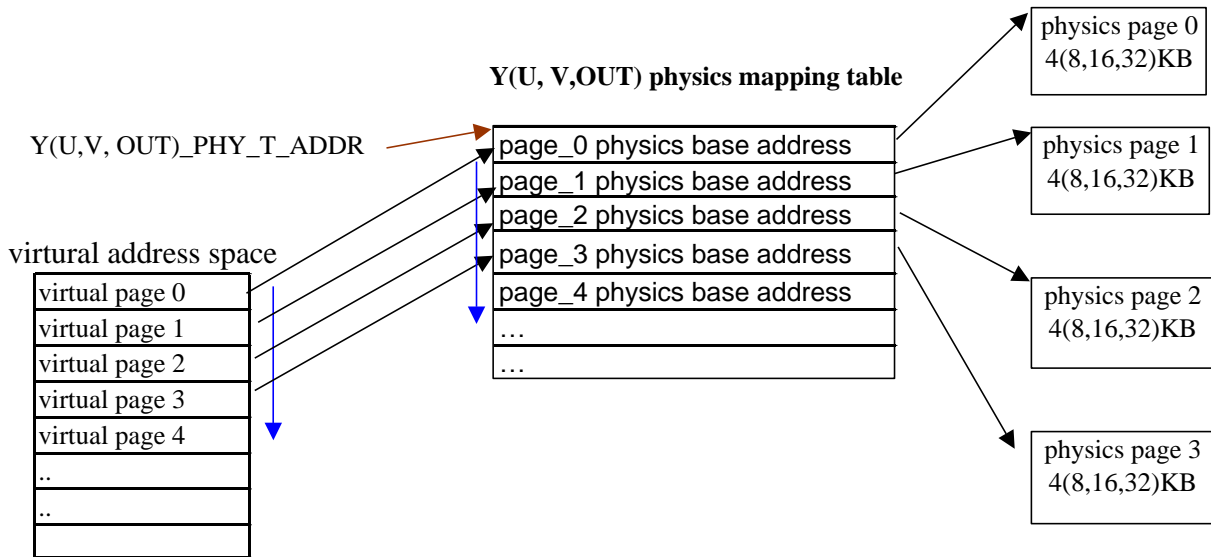
Bits	Name	Description	R/W
31:0	SRC_TLB_ADDR_N	The TLB base address about the next source frame's data which will be DMA in.	R

9.4.17 Destination TLB base address of next frame

DEST_TLB_ADDR_N																0x98																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEST_TLB_ADDR_N																0	0														
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	DEST_TLB_ADDR_N	The TLB base address about the next frame's data that will be DMA out.	R

9.4.18 ADDRESS Mapping



The Y (U, V, OUT)_PHY_T_ADDR should store the **base address** of the Y (U, V, OUT) physics-mapping table. In the Y (U, V, OUT) physics-mapping table, it should contain different physics page base address, and the physics page must be 4(8, 16, 32)KB align.

9.4.19 Input Geometric Size Register

IN_FM_GS																0x18																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved		IN_FM_W														Reserved		IN_FM_H													
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:30	Reserved	Writing has no effect, read as zero.	R
29:16	IN_FM_W	The width of the input frame (unit: byte). Y data width is the same as this value while U/V or Cb/Cr data width should do relatively zoom in according to the source data format. And in the source package pattern , this value should be the Y data width also.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	IN_FM_H	The height of the input frame (unit: byte). Y data width is same as this value while U/V or Cb/Cr data width should do relatively zoom in according to the source data format.	RW

9.4.20 Input Y Data Line Stride Register

Y_STRIDE																0x1c																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												Y_S														0	0				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:20	Reserved	Writing has no effect, read as zero.	R
19:0	Y_S	The line stride of the source Y data in the external memory of separated Frame case or packaged Frame stride(Unit: byte).	RW

NOTE: Y_S should be world align.

9.4.21 Input UV Data Line Stride Register

UV_STRIDE																0x20																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	U_S												0	0	V_S														0	0		
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	R/W
31:16	U_S	The line stride of the source U data in the external memory.	RW
15:0	V_S	The line stride of the source V data in the external memory.	RW

NOTE: U_S and V_S should be word align and unit is byte.

9.4.22 Output Frame Start Address Register

OUT_ADDR																0x24																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	OUT_ADDR																																0	0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Bits	Name	Description	R/W
31:0	OUT_ADDR *1	The output buffer's start address.	RW

NOTES:

- *1: When the IPU_CONTROL.DPAGE_MAP == 1, the OUT_ADDR should be the start virtual address.
- it should be word align.

9.4.23 Output Data Address of next frame Register

OUT_ADDR_N																0x90																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OUT_ADDR_N																0	0														
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	OUT_ADDR *1	The output buffer's start address.	RW

NOTES:

- *1: When the IPU_CONTROL.DPAGE_MAP == 1, the OUT_ADDR should be the start virtual address.
- it should be word align.

9.4.24 Output Frame offset on screen Register

FM_XYOFT																0x78																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				YOFT												Reserved				XOFT											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	YOFT	Vertical offset	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	XOFT	Horizon offset	RW

9.4.25 Output Geometric Size Register

OUT_GS																0x28																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															Reserved																
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31	Reserved	Writing has no effect, read as zero.	R
30:16	OUT_FM_W	The width of the output destination frame (unit: byte).	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	OUT_FM_H	The height of the output destination frame (unit: byte).	RW

NOTES:

- 1 In the package YUV out pattern, the OUT_FM_W should be the **pixel number** in a line.
- 2 In the **RGB** out pattern, the OUT_FM_W should be the **data space** width in the RAM.
- 3 In the out package pattern, the OUT_FM_W **must be** even number, and otherwise IPU will not run.

e.g.

Package YUV out: 480x273 (OUT_FM_W: 480 OUT_FM_H: 273)

RGB 888 (or AAA) out: 480x273 (OUT_FM_W: 480*4 OUT_FM_H: 273)

RGB 555 (or 565) out: 480x273 (OUT_FM_W: 480*2 OUT_FM_H: 273)

9.4.26 Output Data Line Stride Register

OUT_STRIDE																0x2c																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														0	0
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	Reserved	Writing has no effect, read as zero.	R
15:0	OUT_S	The line stride of the destination data buffer in the external memory(Unit: byte).	RW

9.4.27 CSC C0 Coefficient Register

CSC_C0_COEF																0x34																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	Reserved																C0_COEF																	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C0_COEF	The C0 coefficient of the YUV/YCbCr to RGB conversion. C0_COEF = [C0 * 1024 + 0.5].	RW
NOTE:			
R = C0*(Y - LUMA_OF) + C1*(Cr-CHROM_OF)			
G = C0*(Y - LUMA_OF) - C2*(Cb-CHROM_OF) - C3*(Cr-CHROM_OF)			
B = C0*(Y - LUMA_OF) + C4*(Cb-CHROM_OF)			

9.4.28 CSC C1 Coefficient Register

CSC_C1_COEF																0x38																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	Reserved																C1_COEF																	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C1_COEF	The C1 coefficient of the YUV/YCbCr to RGB conversion. C1_COEF = [C1 * 1024 + 0.5].	RW
NOTE:			
R = C0*(Y - LUMA_OF) + C1*(Cr-CHROM_OF)			
G = C0*(Y - LUMA_OF) - C2*(Cb-CHROM_OF) - C3*(Cr-CHROM_OF)			
B = C0*(Y - LUMA_OF) + C4*(Cb-CHROM_OF)			

9.4.29 CSC C2 Coefficient Register

CSC_C2_COEF																0x3C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																C2_COEF															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C2_COEF	The C2 coefficient of the YUV/YCbCr to RGB conversion. C2_COEF = [C2 * 1024 + 0.5].	RW
NOTE:			
R = C0*(Y – LUMA_OF) + C1*(Cr-CHROM_OF)			
G = C0*(Y – LUMA_OF) – C2*(Cb-CHROM_OF) – C3*(Cr-CHROM_OF)			
B = C0*(Y – LUMA_OF) + C4*(Cb-CHROM_OF)			

9.4.30 CSC C3 Coefficient Register

CSC_C3_COEF																0x40																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																C3_COEF															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C3_COEF	The C3 coefficient of the YUV/YCbCr to RGB conversion. C3_COEF = [C3 * 1024 + 0.5].	RW
NOTE:			
R = C0*(Y – LUMA_OF) + C1*(Cr-CHROM_OF)			
G = C0*(Y – LUMA_OF) – C2*(Cb-CHROM_OF) – C3*(Cr-CHROM_OF)			
B = C0*(Y – LUMA_OF) + C4*(Cb-CHROM_OF)			

9.4.31 CSC C4 Coefficient Register

CSC_C4_COEF																0x44																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																C4_COEF															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C4_COEF	The C4 coefficient of the YUV/YCbCr to RGB conversion. C4_COEF = [C4 * 1024 + 0.5].	RW
NOTE: $R = C0*(Y - LUMA_OF) + C1*(Cr-CHROM_OF)$ $G = C0*(Y - LUMA_OF) - C2*(Cb-CHROM_OF) - C3*(Cr-CHROM_OF)$ $B = C0*(Y - LUMA_OF) + C4*(Cb-CHROM_OF)$			

9.4.32 Resize Coefficients Table Index Register

RSZ_COEF_INDEX																0x30																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved											HE_IDX					Reserved							VE_IDX								
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:21	Reserved	Writing has no effect, read as zero.	R
20:16	HE_IDX * ¹	Indicates the end address of the horizontal resize look up table.	RW
15:5	Reserved	Writing has no effect, read as zero.	R
4:0	VE_IDX * ¹	Indicates the end address of the vertical resize look up table.	RW

NOTE: The HE_IDX (VE_IDX) should be the depth of the horizontal (vertical) resize look up table minus 1, and how to get HE_IDX or VE_IDX, please refer to 3.34.

9.4.33 Horizontal Resize Coefficients Look Up Table Register group

HRSZ_COEF_LUT (bi-cube)																0x48																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

	Reserved					SIGN	W_COEF_31											SIGN	W_COEF_20											HRSZ_OFT					H_CONF
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:28	Reserve	Writing has no effect, read as zero.	W
27:17	W_COEF_31	Weighting coefficients for pix_3 or pix_1.	W
16:6	W_COEF_20	Weighting coefficients for pix_2 or pix_0.	W
5:1	HRSZ_OFT	Horizontal Resize pixel offset.	W
0	H_CONF	Start to configure the horizontal resize table, read as zero: 1: start.	W

HRSZ_COEF_LUT (bilinear)																0x48																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved															SIGN	W_COEF0											H_OFT					H_CONF
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:17	Reserve	Writing has no effect, read as zero.	R
16:6	W_COEF0	Weighting coefficients for pix_0.	W
5:1	H_OFT	Horizontal Resize pixel offset.	W
0	H_CONF	Start to configure the horizontal resize table, read as zero: 1: start.	W

NOTES:

- The **bilinear zooming** weighting coefficients should be calculated as following: because it is 11 bits length, a one bit is the sign-bit, so, that is to say the precision is 1/512.

For up-scaling,

$$W_k = 1 - (k*n/m - [k*n/m]), k = 0, 1, \dots m-1.$$

For down-scaling,

for (t=0, k=0; k < n; k++)

{

 If $((t*n+1)/m) - k \geq 1$ { $W_k = 0$;

 else if $((t*n+1)/m - k == 0)$ { $W_k = 1$; t++;}

 else { $W_k = 1 - ((t*n+1)/m - [t*n/m])$; t++;}

}

$W_COEF_k = [512 * W_k]$ (stands for get the rounding integer, $[20.33] = 20$ while $[20.66] = 21$)

Here n stands for original pixel points, m stands for pixel points after resize. For example down-scaling 5:3, $n = 5$, $m = 3$. Moreover, m and n are prime, that is, for example 8:2 should be converted to 4:1.

When IPU_CONTROL.RSZ_EN set as 1 and $m:n = 1:1$, all coefficients should be calculated as up-scale case.

- 2 The **bi-cube zooming** weighting coefficients should be calculate as following: because it is 10 bits length, that is to say the precision is $1/512$.

Step 1:

For up-scaling,

$$W_k = 1 - (k*n/m - [k*n/m]), k = 0, 1, \dots m-1.$$

For down-scaling,

for (t=0, k=0; k < n; k++)

{

 if ((t*n+1)/m - k >= 1) { $W_k = 0$;

 else if ((t*n+1)/m - k == 0) { $W_k = 1$; t++;}

 else { $W_k = 1 - ((t*n+1)/m - [t*n/m])$; t++;}

}

$W_COEF_k = [512 * W_k]$ (stands for get the rounding integer, $[20.33] = 20$ while $[20.66] = 21$)

Here n stands for original pixel points, m stands for pixel points after resize. For example down-scaling 5:3, $n = 5$, $m = 3$. Moreover, m and n are prime, that is, for example 8:2 should be converted to 4:1.

When IPU_CONTROL.RSZ_EN set as 1 and $m:n = 1:1$, all coefficients should be calculated as up-scale case.

Step 2:

After calculate the W_k , then using the following rule to get the coefficients:

double SinXDivX(double x)

{

 const float a = -1; //a can be a=-2,-1,-0.75,-0.5 and so on to control the blur level

 double x2=x*x;

 double x3=x2*x;

 if (x<=1)

 return (a+2)*x3 - (a+3)*x2 + 1;

 else if (x<=2)

 return a*x3 - (5*a)*x2 + (8*a)*x - (4*a);

 else

 return 0;

}

$W0 = 512 * \text{SinXDivX}(1+W_k)$

$W1 = 512 * \text{SinXDivX}(W_k)$

$W2 = 512 * \text{SinXDivX}(1-W_k)$

$W3 = 512 * \text{SinXDivX}(2-W_k)$

Step 3:

And then the zooming weight coefficient should set to IPU as following:

Prepare: Set H_CONF to 1

- a set W_{4n+1} & W_{4n+0}
- b set W_{4n+3} & W_{4n+2}

Index(n)	step	W0	W1	step	W2	W3
0	0	-34	129	1	100	-19
1	2	-45	45	3	77	-33
2	4	-12	122	5	94	-56
3	6	-13	230	7	123	-77
4	8	-23	11	9	45	-100
5	10	-19	87	11	69	-90
6	12	-12	79	13	148	-8

3 Calculate the H_OFT.

Step 1: calculate the line in enable flag (IN_EN) and out enable flag (OUT_EN) table.

IN_EN: In down scale case, IN_EN always equals 1.

In up scale case,

For (i=0, k=0; k < m; k++) {

If (i <= k*n/m) { IN_EN [k] = 1; i++;}

else { IN_EN [k] = 0;}

OUT_EN: In up scale case, OUT_EN always equals 1.

In down scale case,

For (t=0, k=0; k < n; k++) {

If ([(t*n+1)/m] - k >= 1)

OUT_EN [k] = 0;

else {OUT_EN [k] = 1; t++;}

}

Step 2: calculate the H_MAX_LUT.

$H_MAX_LUT = \max(m, n) - 1$

Step 3: Calculate the LUT.

```
int hoft_table_buf[33];
int hcoef_table_buf[33];
int voft_table_buf[33];
int vcoef_table_buf[33];
int *hoft_table = &hoft_table_buf[1];
int *hcoef_table = &hcoef_table_buf[1];
int *voft_table = &voft_table_buf[1];
int *vcoef_table = &vcoef_table_buf[1];
int in_ofst_tmp = 0;
int hcoef_real_height = 0 ;
int vcoef_real_height = 0 ;
int coef_tmp = 0 ;
j = -1 ;
for (i=0; i<=H_MAX_LUT+1; i++)
```

```

    {
        if ( h_lut[i].out_n )
        {
            hoft_table[j] = (h_lut[i].in_n == 0)? 0: in_ofst_tmp;
            hcoef_table[j] = coef_tmp;
            coef_tmp = h_lut[i].coef;
            in_ofst_tmp = h_lut[i].in_n==0? in_ofst_tmp : h_lut[i].in_n ;
            j++;
        }
        else
            in_ofst_tmp = h_lut[i].in_n + in_ofst_tmp;
    }
    if ( h_lut[0].out_n )
    {
        hoft_table[j] = (h_lut[0].in_n == 0)? 0: in_ofst_tmp;
        hcoef_table[j] = coef_tmp;
    }
    j++;
    hcoef_real_heiht = j;
    RSZ_COEF_INDEX. HE_IDX = j -1;

```

Step 4: Calculate the last table of resize coefficient.

Bilinear:

```
for (cnt =0 ; cnt<hcoef_real_heiht ; cnt ++)
```

```
    HRSZ_COEF_LUT_bilinear[cnt] = (hcoef_table[cnt], hoft_table[cnt]);
```

Bicube:

```
int sinxdivx_table_8[(2<<9)+1];
```

```
for ( i = 0 ; i < (2<<9); ++i)
```

```
{
    sinxdivx_table_8[i] = (int)(0.5 + 512*sinxdivx(i*(1.0/512)));
}
```

```
int u_8;
```

```
for (i = 0 ; i <hcoef_real_heiht; i++)
```

```
{
    int au_8[4];
    u_8 = 512 - hcoef_table[i];
    cube_hcoef_table[i][0] = sinxdivx_table_8[(1<<9)+u_8];
    cube_hcoef_table[i][1] = sinxdivx_table_8[u_8];
    cube_hcoef_table[i][2] = sinxdivx_table_8[(1<<9)-u_8];
    cube_hcoef_table[i][3] = sinxdivx_table_8[(2<<9)-u_8];
}
```

```
for (cnt =0 ; cnt<hcoef_real_heiht ; cnt ++)
```

```
    HRSZ_COEF_LUT_bicube[cnt] = (cube_hcoef_table[cnt], hoft_table[cnt]);
```

4 Following are two examples of setting LUT in bilinear scale mode.

5

Resize coefficients for 7:3:

W	W_COEF	IN_EN	OUT_EN	Pixel 1	Pixel 2	OUT
2/3	341	1	1	P [0]	P [1]	$P [0] * 2/3 + P [1] * 1/3$
0	0	1	0	P [1]	P [2]	No new pixel out
1/3	171	1	1	P [2]	P [3]	$P [2] * 1/3 + P [3] * 2/3$
0	0	1	0	P [3]	P [4]	No new pixel out
0	0	1	0	P [4]	P [5]	No new pixel out
1	512	1	1	P [5]	P [6]	$P [5] * 1 + P [6] * 0$
0	0	1	0	P [6]	P [7]	No new pixel out

Parameter set to IPU is following:

index	W	W_COEF	OFSE T	Pixel1	Pixel 2	OUT
0	2/3	341	2	P [0]	P [1]	$P [0] * 2/3 + P [1] * 1/3$
1	1/3	171	3	P [2]	P [3]	$P [2] * 1/3 + P [3] * 2/3$
2	1	512	2	P [5]	P [6]	$P [5] * 1 + P [6] * 0$

Resize coefficients for 3:5:

W	W_COEF	IN_EN	OUT_EN	Pixel 1	Pixel 2	OUT
1	512	1	1	P [0]	P [1]	$P [0] * 1 + P [1] * 0$
2/5	205	0	1	P [0]	P [1]	$P [0] * 2/5 + P [1] * 3/5$
4/5	410	1	1	P [1]	P [2]	$P [1] * 4/5 + P [2] * 1/5$
1/5	102	0	1	P [1]	P [2]	$P [1] * 1/5 + P [2] * 4/5$
3/5	307	1	1	P [2]	P [3]	$P [2] * 3/5 + P [3] * 2/5$

The parameter set to IPU is as following:

index	W	W_COEF	OFFSET	Pixel 1	Pixel 2	OUT
0	1	512	0	P [0]	P [1]	$P [0] * 1 + P [1] * 0$
1	2/5	205	1	P [0]	P [1]	$P [0] * 2/5 + P [1] * 3/5$
2	4/5	410	0	P [1]	P [2]	$P [1] * 4/5 + P [2] * 1/5$
3	1/5	102	1	P [1]	P [2]	$P [1] * 1/5 + P [2] * 4/5$
4	3/5	307	1	P [2]	P [3]	$P [2] * 3/5 + P [3] * 2/5$

9.4.34 Vertical Resize Coefficients Look Up Table Register group

VRSZ_COEF_LUT (bi-cube)																0x4C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				SIGN	W_COEF_31						SIGN	W_COEF_20						VRSZ_OFT				V_CONF									
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:28	Reserved	Writing has no effect, read as zero.	W
27:17	W_COEF_31	Weighting coefficients for pix_3 or pix_1.	W
16:6	W_COEF_20	Weighting coefficients for pix_2 or pix_0.	W
5:1	VRSZ_OFT	Vertical Resize pixel offset.	W
0	V_CONF	Start to configure the vertical resize table, read as zero: 1: start.	W

VRSZ_COEF_LUT (bilinear)																0x4C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																W_COEF0						V_OFT				V_CONF					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:17	Reserved	Writing has no effect, read as zero.	R
16:6	W_COEF0	Weighting coefficients for pix_0.	W
5:1	V_OFT	Vertical Resize pixel offset.	W
0	V_CONF	Start to configure the vertical resize table, read as zero: 1: start.	W

NOTE: refer to Horizontal HRSZ_COEF_LUT.

9.4.35 Calculation for Resized width and height

For software, to preset correct value for register OUT_GS, please refer to following formula.

Set IW stand for original input frame width, IH stand for original input frame height, OW stand for new frame width after resize, OH stand for new frame height after resize.

In Up-scale case ($n < m$):

If $[(IW - 1) * (m/n)] * (n/m) == (IW-1)$ then

$OW = [(IW - 1) * (m/n)] + 1;$

Else $OW = [(IW - 1) * (m/n)] + 2;$

If $[(IH - 1) * (m/n)] * (n/m) == (IH-1)$ then
 $OH = [(IH - 1) * (m/n)] + 1;$
 Else $OH = [(IH - 1) * (m/n)] + 2;$

In Down-scale case ($n>m$):

If $[(IW - 1) * (m/n)] * (n/m) == (IW-1)$ then
 $OW = [(IW - 1) * (m/n)];$
 Else $OW = [(IW - 1) * (m/n)] + 1;$
 If $[(IH - 1) * (m/n)] * (n/m) == (IH-1)$ then
 $OH = [(IH - 1) * (m/n)];$
 Else $OH = [(IH - 1) * (m/n)] + 1;$

For example:

A 36x46 frame with the horizontal resize ratio of 4:5 (up-scale) and vertical resize ratio of 7:6 (down-scale), by the expressions above we get its new size after resize from the following process.

For Width: $[(36 - 1) * (5/4)] * (4/5) = 34.4 \neq (36-1)$
 So $OW = [(36 - 1) * (5/4)] + 2 = 45$
 For Height: $[(46 - 1) * (6/7)] * (7/6) = 44.33 \neq (46 - 1)$
 So $OH = [(46 - 1) * (6/7)] + 1 = 39$

9.4.36 CSC Offset Parameter Register

CSC_OFFSET_PARA																0x50																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								CHROM_OF								Reserved				LUMA_OF											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

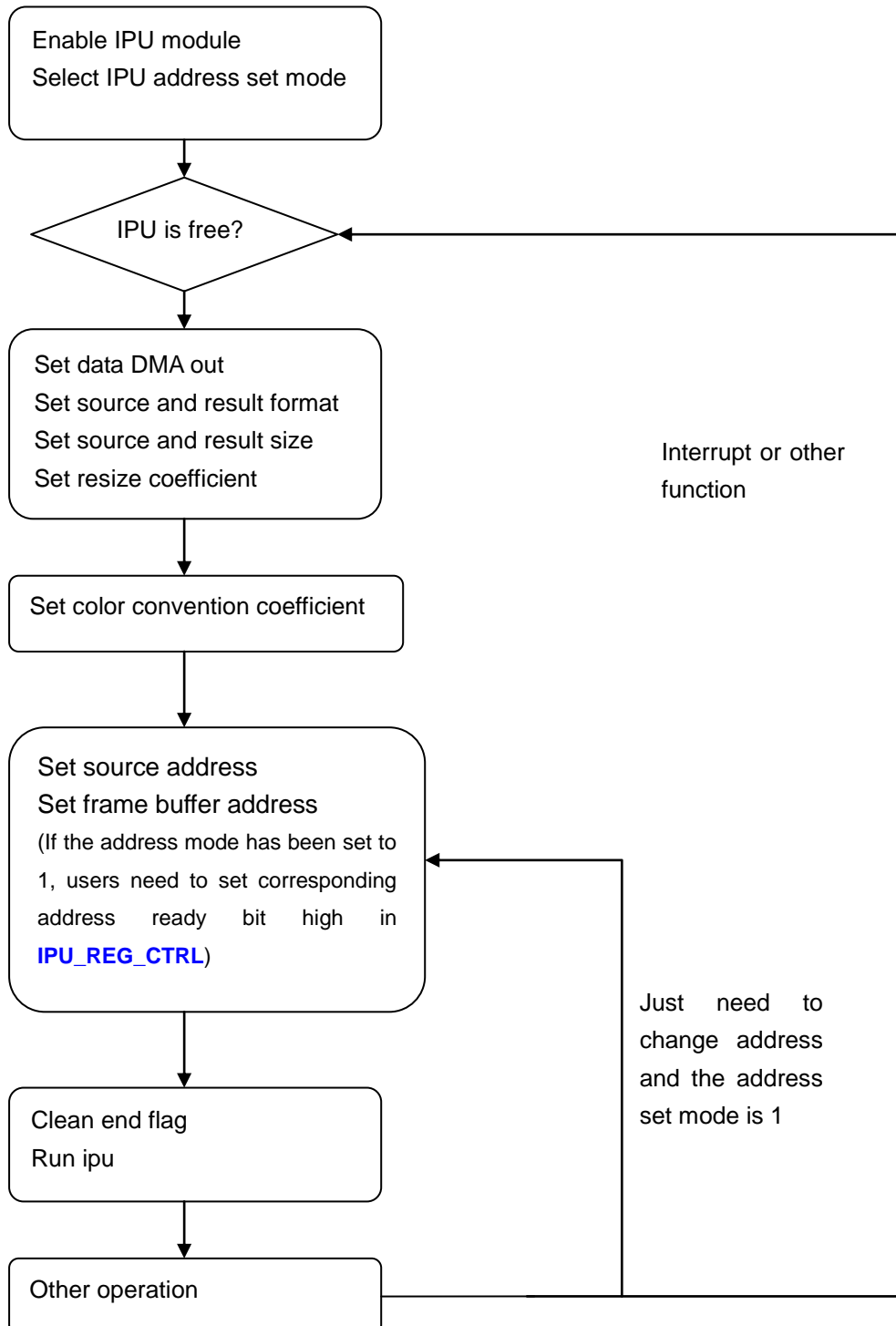
Bits	Name	Description	R/W
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	CHROM_OF	Chroma offset value.	RW
15:8	Reserved	Writing has no effect, read as zero.	R
7:0	LUMA_OF	Luma offset value.	RW

NOTE:

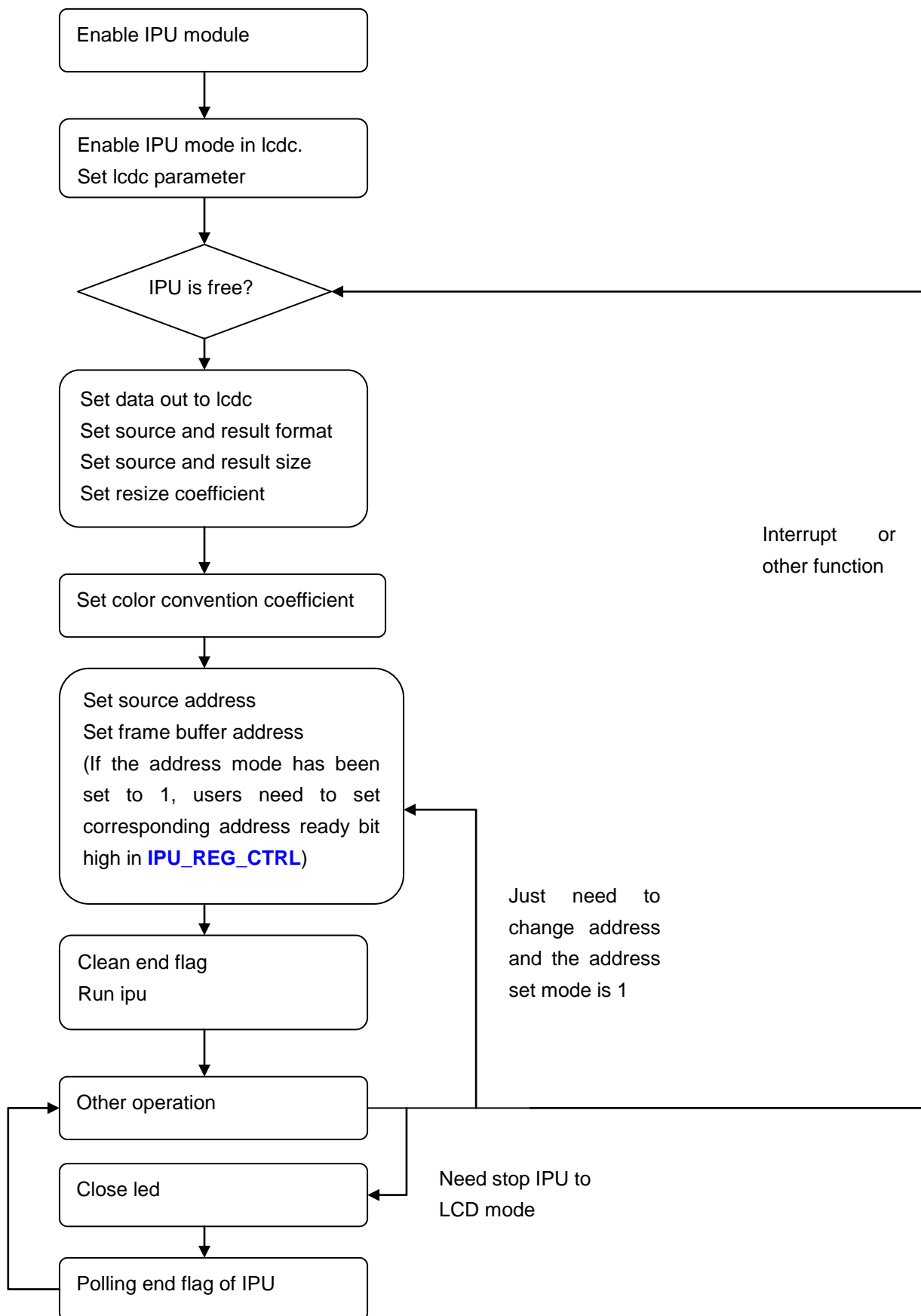
$R = C0*(Y - LUMA_OF) + C1*(Cr-CHROM_OF)$
 $G = C0*(Y - LUMA_OF) - C2*(Cb-CHROM_OF) - C3*(Cr-CHROM_OF)$
 $B = C0*(Y - LUMA_OF) + C4*(Cb-CHROM_OF)$

9.5 IPU Operation Flow

9.5.1 Data out to frame buffer



9.5.2 Data out to lcdc



9.5.3 Operation example

Table 9-2 no mapping mode

Step	Action
Base	Chip_enable()
Base_0	ipu_addr_sel(1);
0	Do { } while {!polling_end_flag}
1	set_primary_ctrl(VRSZ_ENABLE, HRSZ_ENABLE, CSC_EN, irq_en); //
2	set_source_ctrl(source_pkg_sel, SPAGE_SEL);
3	set_out_ctrl(lcdc_sel, DPAGE_SEL, DISP_SEL, FIELD_SEL, FIELD_CONF_EN);
4	set_scale_ctrl(V_SCALE, H_SCALE);
5	set_ipu_fmt(RGB888_OUT_FMT, OUT_OFT_RGB, OUT_FMT, OUT_Y1UY0V , IN_OF_YUYV, IN_FM_YUV444);
6	set_inframe_gsize(FIN_W, FIN_H, FIN_Y_STRIDE, FIN_U_STRIDE, FIN_V_STRIDE);
7	set_y_addr((unsigned int)fin_y & 0x1FFFFFFF); set_u_addr((unsigned int)fin_y & 0x1FFFFFFF); set_v_addr((unsigned int)fin_y & 0x1FFFFFFF);
8	set_outframe_gsize(FOUT_W, FOUT_H , FOUT_STRIDE);
9	set_out_addr((unsigned int)fout & 0x00000FFF);
9A	set_addr_ready(0xFF); NOTE: this step is necessary when ipu address set mode is 1.
10	set_csc_c0(YUV_CSC_C0); set_csc_c1(YUV_CSC_C1); set_csc_c2(YUV_CSC_C2); set_csc_c3(YUV_CSC_C3); set_csc_c4(YUV_CSC_C4);
11	set_csc_offset_para (128, 0);
12	set_rsz_lut_end(H_MAX_LUT-1, V_MAX_LUT-1);
13	start_hlut_coef_write(); NOTE: This step is necessary before write new LUT.
14	for (i=0;i<H_MAX_LUT;i++) { set_hrsz_lut_coef(h_lut[i].coef, h_lut[i].in_n, h_lut[i].out_n); }
15	start_vlut_coef_write(); NOTE: This step is necessary before write new LUT.
16	for (i=0;i<V_MAX_LUT;i++) { set_vrsz_lut_coef(v_lut[i].coef, v_lut[i].in_n, v_lut[i].out_n); }
17	Clean_end_flag(); run_ipu();

Table 9-3 mapping mode

Step	Action
Prepare	<pre> y_phy_table[0] = ((unsigned int)fin_y & 0x0FFFF000) 0x20000000 ; u_phy_table[0] = ((unsigned int)fin_u & 0x0FFFF000) 0x20000000 ; v_phy_table[0] = ((unsigned int)fin_v & 0x0FFFF000) 0x20000000 ; out_phy_table[0] = ((unsigned int)fout & 0x0FFFF000) 0x20000000 ; for (i =1; i<100; i++){ y_phy_table[i] = y_phy_table[i-1] + 4096 ; u_phy_table[i] = u_phy_table[i-1] + 4096 ; v_phy_table[i] = v_phy_table[i-1] + 4096 ; out_phy_table[i] = out_phy_table[i-1] + 4096 ; } </pre>
Base	Chip_enable()
Base_0	ipu_addr_sel(1);
0	Do { } while {!polling_end_flag}
1	set_primary_ctrl(VRSZ_ENABLE, HRSZ_ENABLE, CSC_EN, irq_en) ;//
2	set_source_ctrl(source_pkg_sel, SPAGE_SEL) ;
3	set_out_ctrl(lcdc_sel, DPAGE_SEL, DISP_SEL, FIELD_SEL, FIELD_CONF_EN) ;
4	set_scale_ctrl(V_SCALE, H_SCALE) ;
5	set_ipu_fmt(RGB888_OUT_FMT, OUT_OFT_RGB, OUT_FMT, OUT_Y1UY0V , IN_OF_YUYV, IN_FM_YUV444) ;
6	set_inframe_gsize(FIN_W, FIN_H, FIN_Y_STRIDE, FIN_U_STRIDE, FIN_V_STRIDE) ;
7	set_y_addr((unsigned int)fin_y & 0xFFF); set_u_addr((unsigned int)fin_y & 0xFFF); set_v_addr((unsigned int)fin_y & 0xFFF);
8	set_outframe_gsize(FOUT_W, FOUT_H , FOUT_STRIDE);
9	set_out_addr((unsigned int)fout & 0x0000FFF);
10	set_y_phy_t_addr((unsigned int)y_phy_table & 0x1FFFFFFF) ; set_u_phy_t_addr((unsigned int)u_phy_table & 0x1FFFFFFF) ; set_v_phy_t_addr((unsigned int)v_phy_table & 0x1FFFFFFF) ; set_out_phy_t_addr((unsigned int)out_phy_table & 0x1FFFFFFF) ;
10A	set_addr_ready(0xFF); NOTE: this step is necessary when ipu address set mode is 1.
11	set_csc_c0(YUV_CSC_C0); set_csc_c1(YUV_CSC_C1); set_csc_c2(YUV_CSC_C2); set_csc_c3(YUV_CSC_C3); set_csc_c4(YUV_CSC_C4);
12	set_csc_ofset_para (128, 0) ;
13	set_rsz_lut_end(H_MAX_LUT-1, V_MAX_LUT-1);
14	start_hlut_coef_write(); NOTE: This step is necessary before write new LUT.

```

15     for (i=0;i<H_MAX_LUT;i++) {
           set_hrsz_lut_coef(h_lut[i].coef, h_lut[i].in_n, h_lut[i].out_n);
       }
16     start_vlut_coef_write();
       NOTE: This step is necessary before write new LUT.
17     for (i=0;i<V_MAX_LUT;i++) {
           set_vrsz_lut_coef(v_lut[i].coef, v_lut[i].in_n, v_lut[i].out_n);
       }
18     Clean_end_flag();
       run_ipu();
    
```

9.6 Special Instruction

9.6.1 Resizing size feature

Input size (W x H)		Output size (W x H)	
Min	4x4	Disable vertical scale	Min: 4x4
			Max: 4095x4095
Max	4095x4095	Enable vertical scale	Min: 4x4
			Max: 1280x4095

9.6.2 Color convention feature


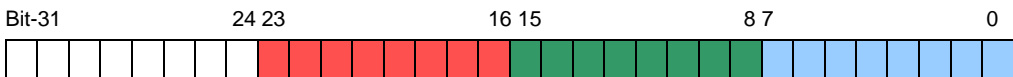
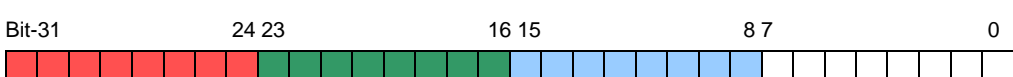
Source format	Output format	Parameter configure (necessary)
RGB	RGB	IPU_CONTROL.CSC_EN =0
		IPU_CONTROL.SPKG_SEL = 0 or 1
		D_FMT.IN_FMT
		D_FMT.OUT_FMT = 2'b00, 2'b01, 2'b10
YUV	RGB	IPU_CONTROL.CSC_EN =1
		IPU_CONTROL.SPKG_SEL
		D_FMT.IN_FMT
		D_FMT.IN_OFT (<i>IPU_CONTROL.SPKG_SEL == 1</i>)
		D_FMT.OUT_FMT = 2'b00, 2'b01, 2'b10
		D_FMT.RGB_OUT_OFT.
		CSC_C0 (1,2,3,4)_COEF, CSC_OFFSET_PARA
YUV	YUV (package)	IPU_CONTROL.CSC_EN =0
		IPU_CONTROL.SPKG_SEL
		D_FMT.IN_FMT
		D_FMT.IN_OFT (<i>IPU_CONTROL.SPKG_SEL == 1</i>)


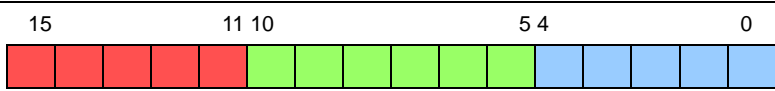
	D_FMT.OUT_FMT = 2'b11
--	-----------------------

9.6.3 YUV/YCbCr to RGB CSC Equations

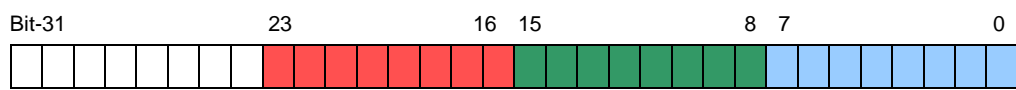
Input data	Matrix	CSC_COEF
YUV	$R = C0*(Y - X0) + C1*(V-128)$	CSC_C0_COEF = 0x400
	$G = C0*(Y - X0) - C2*(U-128) - C3*(V-128)$	CSC_C1_COEF = 0x59C
	$B = C0*(Y - X0) + C4*(U-128)$	CSC_C2_COEF = 0x161
	X0: 0	CSC_C3_COEF = 0x2DC
	C0: 1	CSC_C4_COEF = 0x718
	C1: 1.4026	
	C2: 0.3444	
YCbCr	$R = C0*(Y - X0) + C1*(Cr-128)$	CSC_C0_COEF = 0x4A8
	$G = C0*(Y - X0) - C2*(Cb-128) - C3*(Cr-128)$	CSC_C1_COEF = 0x662
	$B = C0*(Y - X0) + C4*(Cb-128)$	CSC_C2_COEF = 0x191
	X0: 16	CSC_C3_COEF = 0x341
	C0: 1.164	CSC_C4_COEF = 0x811
	C1: 1.596	
	C2: 0.391	
	C3: 0.813	
	C4: 2.018	

9.6.4 Output data package format (RGB order)

Format	Package
RGBAAA	 <p>Bit-31 29 20 19 10 9 0</p> <p>EMPTY</p>
	 <p>Bit-31 24 23 16 15 8 7 0</p> <p>EMPTY</p> <p style="text-align: center;">Or</p>  <p>Bit-31 24 23 16 15 8 7 0</p> <p>EMPTY</p>

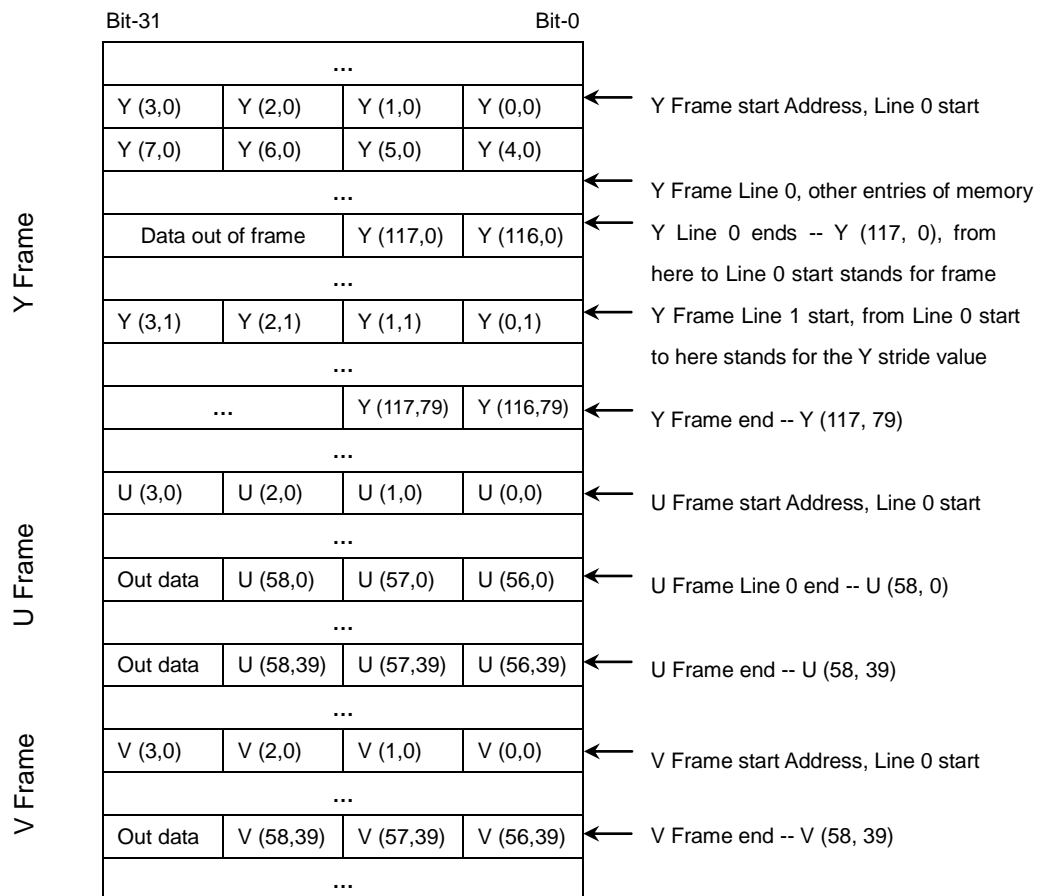
RGB555	
RGB565	
NOTE: All R/G/B data are little-endian type; all the empty bits in the above figure are filled with 0.	

9.6.5 Input data package format (RGB order)

Format	Package
RGBPACK	

9.6.6 Source Data storing format in external memory (separated YUV Frame)

Example: YUV420 118x80 frame



NOTES:

- 1 Every line's start address should be word aligned.
- 2 All pixel data should be stored as little-endian format.
- 3 Destination data (RGB) storing format in external memory is similar with above figure, but RGB555 and RGB565 frame's every line start address can be half-word aligned. (RGB888 frame still need word aligned)

10 LVDS Controller

10.1 Overview

This product is a single-Link high speed LVDS (Low-Voltage Differential Signaling) transmitter used for digital flat panel display systems. It's compatible with ANSI/TIA/EIA-644-A (LVDS) Standard. The transmitter converts 28bits parallel TTL data into four LVDS data streams. An in-phase transmit clock is transmitted in parallel with the data streams over a fifth LVDS link. It support full HDTV display up to 1920x1080p @ 60 Hz.

Feature:

- 25 to 150 MHz input clock support
- Supports VGA, SVGA, XGA , SXGA and HDTV
- Compatible with TIA/EIA-644 LVDS standard
- Support 24-bit Flat Panel Display
- Support VESA and JEIDA LVDS Data format

10.2 Register Description

In this section, we will describe the registers in LVDS controller. Following table lists all the register definitions. All registers' 32bit addresses are physical addresses. And detailed function of each register will be described below.

Table 10-1 LVDS Register Description

Name	Description	RW	Reset Value	Address	Access Size
TXCTRL	LVDS Transmitter 's Control Register	RW	0x80040060	0x130A03C0	32
TXPLL0	LVDS Transmitter's PLL Control Register 0	RW	0x60001304	0x130A03C4	32
TXPLL1	LVDS Transmitter's PLL Control Register 1	RW	0x61000000	0x130A03C8	32
TXECTRL	LVDS Transmitter's Enhance Control Register	RW	0x00000030	0x130A03CC	32

10.2.1 TXCTRL (LVDS Transmitter Control Register)

The register TXCTRL is used to control LVDS to work.

TXCTRL																0x130A03C0																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVDS_MODEL_SEL	TX_PDB	TX_PDB_CK	Reserved	RESERVE_7	RESERVE_6	RESERVE_5	RESERVE_4	RESERVE_3	RESERVE_2	RESERVE_1	RESERVE_0	Reserved	TX_RSTB	TX_CKBIT	TX_CKBYTE				TX_CKOUT_PHA_S	TX_CKOUT_SET	TX_OUT_SEL		TX_DLY_SEL	TX_AMP_ADJ	TX_LVDS						
RST	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Bits	Name	Description	RW
31	LVDS_MODEL_SEL	VESA model or JEIDA model select. 0: JEIDA; 1:VESA.	RW
30	TX_PDB	Data channel Power down. 0: power down.	RW
29	TX_PDB_CK	Clock channel Power down. 0: power down.	RW
28	Reserved	Writing has no effect, read as zero.	R
27	LVDS_RESERVE_7	Reserved.	RW
26	LVDS_RESERVE_6	Reserved.	RW
25	LVDS_RESERVE_5	Reserved.	RW
24	LVDS_RESERVE_4	Reserved.	RW
23	LVDS_RESERVE_3	Reserved.	RW
22	LVDS_RESERVE_2	Reserved.	RW
21	LVDS_RESERVE_1	Reserved.	RW
20	LVDS_RESERVE_0	Reserved.	RW
19	Reserved	Writing has no effect, read as zero.	R
18	TX_RSTB	System reset signal. 0: Reset.	RW
17	TX_CKBIT_PHA_SEL	7x clock sampling edge configuration. 0: Rising edge; 1: Falling Edge.	RW
16	TX_CKBYTE_PHA_SEL	1x clock sampling edge configuration. 0: Rising edge; 1:Falling Edge.	RW
15:13	TX_CKOUT_PHA_S	Output data start-edge tuning in 1x clock output mode. 000: 0 of T7X 001: 1 of T7X ... 111: 7 of T7X	RW
12	TX_CKOUT_SET	TX clock channel output clock frequency set. 0: 1x clock output 1: 7x clock output	RW
11	TX_OUT_SEL	Transmitter output select. 0: LVDS output; 1: CMOS RGB output.	RW
10:8	TX_DLY_SEL	Input clock edge delay control, for setup/hold time fine tuning.	RW
7	TX_AMP_ADJ	LVDS output swing control.	RW

		When AMP_ADJ=1, LVDS output swing is adjustable by CR<2:0>.	
6	TX_LVDS	Output amplitude tuning in mode of 'TX_AMP_ADJ = 0. 0: VOD=200mV; 1: VOD=350mV.	RW
5:3	TX_CR	Digital logic input used to control output swing level.	RW
2	TX_CR_CK	Additional control bit of output swing level.	RW
1	TX_OD_S	Output level selectable pin.	RW
0	TX_OD_EN	Tx output control functions. 0: Disable ;1: Enable.	RW

10.2.2 TXPLL0 (LVDS Transmitter's PLL Control Register 0)

The register TXPLL0 is used to control PLL to work.

TXPLL0																0x130A03C4																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	LVDS_PLL_LOCK	PLL_EN	BG_PWD	Reserved	PLL_SSC_EN	PLL_SSC_MODE	PLL_TEST	Reserved		PLL_POST_DIVA		PLL_POST_DIVB						Reserved	PLL_PLLN						PLL_TEST_DIV	PLL_IN_BYPASS	PLL_INDIV						
RST	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0

Bits	Name	Description	RW
31	LVDS_PLL_LOCK	Lock detection output. 1: Lock.	R
30	PLL_EN	PLL enable control. 1: enable.	RW
29	BG_PWD	Band-gap power down control. 1: power down.	RW
28	Reserved	Writing has no effect, read as zero.	R
27	PLL_SSC_EN	SSC function enable control. 1: Enable.	RW
26	PLL_SSC_MODE	SSC mode select. 0: Down spread 1: Center spread	RW
25	PLL_TEST	Test enable control. 1:Enable.	RW
24:23	Reserved	Writing has no effect, read as zero.	R
22:21	PLL_POST_DIVA	Post divider control bits A.	RW
20:16	PLL_POST_DIVB	N/C.	RW
15	Reserved	Writing has no effect, read as zero.	R
14:8	PLL_PLLN[6:0]	PLL feedback divider value configure.	RW
7:6	PLL_TEST_DIV	Output divider ratio control in test mode. 00: 1/2	RW

		01: 1/4 10: 1/8 11: 1/16	
5	PLL_IN_BYPASS	Input divider bypass.	RW
4:0	PLL_INDIV	Input divider value configure.	RW

10.2.3 TXPLL1 (LVDS Transmitter's PLL Control Register 0)

The register TXPLL1 is used to control PLL to work.

	TXPLL1																0x130A03C8															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PLL_ICP_SEL			Reserved	PLL_KVCO		PLL_IVCO_SEL		PLL_SSCN								PLL_COUNT								PLL_GAIN							
RST	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:29	PLL_ICP_SEL	Charge-pump current configure.	RW
28	Reserved	Writing has no effect, read as zero.	R
27:26	PLL_KVCO	VCO gain control.	RW
25:24	PLL_IVCO_SEL	VCO biasing current setup.	RW
23:17	PLL_SSCN	Internal divider for 30KHz clock generation.	RW
16:4	PLL_COUNT	SSC counter.	RW
3:0	PLL_GAIN	SSC counter gain.	RW

10.2.4 TXCTRL (LVDS Transmitter's Enhance Control)

The register TXCTRL is used to control LVDS.

	TXCTRL																0x130A03CC															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																TX_EM_S	TX_EM_EN	Reserved	LDO_VO_S	PLL_PI_BP	CK_PHA_FINE	CK_PHA_COAR									
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bits	Name	Description	RW
31:11	Reserved	Writing has no effect, read as zero.	R
10:9	TX_EM_S	Emphasis level configure	RW

8	TX_EM_EN	Emphasis driver control	RW
7	Reserved	Writing has no effect, read as zero.	R
6:5	TX_LDO_VO_S	Internal LDO output voltage configure 00: 1.0V 01: 1.1V 10: 1.2V 11: 1.3V	RW
4	PLL_PI_BP	PLL phase interpolator bypass. 1: Bypass	RW
3:2	TX_CK_PHA_FINE	Internal 7x clock phase fine turning for data setup/hold time optimization.	RW
1:0	TX_CK_PHA_COAR	Internal 7x clock phase coarse tuning for data setup/hold time optimization.	RW

10.3 Operate Description

10.3.1 Operate Mode

Table 10-2 Operation Mode Description

Work Setting	LVDS OUTPUT		CMOS OUTPUT	OUTPUT HI-Z	OUTPUT 0	Power Down
	7X CLK OUT	1X CLK OUT				
BG_PWD	0	0	0	X	X	1
PLL_EN	1	1	1	X	X	0
TX_PDB	1	1	1	1	1	0
TX_PDB_CK	1	1	1	1	1	0
TX_OD_EN	1	1	1	0	0	0
TX_OD_S	0	0	1	0	1	0
TX_OUT_SEL	0	0	1	X	X	X
TX_CKOUT_SET	1	0	X	X	X	X

Note: When in power down mode, the LVDS output is Hi-Z state, and PLL_LOCK=0.

10.3.2 LVDS Output Amplitude Control

Table 10-3 LVDS Output Amplitude Control

TX_AMP_ADJ	TX_LVDS	Output Voltage
0	0	200mV
	1	350mV
1	X	controlled by TX_CR[2:0] and TX_CR_CK

Table 10-4 LVDS Output Swing Level Control

TX_CR_CK	TX_CR[2]	TX_CR[1]	TX_CR[0]	I _{OUT} (mA)	V _{OD} (mV)
0	0	0	0	1.5	150
	0	0	1	2.0	200
	0	1	0	2.5	250
	0	1	1	3.0	300
	1	0	0	3.5	350
	1	0	1	4.0	400
	1	1	0	5.0	500
	1	1	1	6.0	600
1	0	0	0	6.5	650
	0	0	1	7.0	700
	0	1	0	7.5	750
	0	1	1	8.0	800
	1	0	0	8.5	850
	1	0	1	9.0	900
	1	1	0	10	1000
	1	1	1	11	1100

Table 10-5 LVDS Output Level Function

TX_OD_S	TX_OD_EN	Output
1	1	CMOS(RGB Data)
0	1	LVDS (LVDS Data)
1	0	0
0	0	HI-Z

10.3.3 LVDS Input Clock Edge-Delay Control

Table 10-6 LVDS_TX Clock Edge Delay Control

TX_DLY_SEL[2]	TX_DLY_SEL[1]	TX_DLY_SEL[0]	Edge Delay (ns)
0	0	0	0.1
0	0	1	0.2
0	1	0	0.5
0	1	1	1
1	0	0	1.5
1	0	1	2.0
1	1	0	2.5
1	1	1	3.0

Note: TX_DLY_SEL is used to tune the PLL output 7x clock edge for better data sampling operation.

10.3.4 Output data start-edge Control

To work with different type of LVDS receiver, the LVDS transmitter can be configured to output 7x or 1x clock by TX_CKOUT_SET.

In 1x clock output application, the clock sampling edge refer to the data start-edge may have effect on the receiver's operation. So TX_CKOUT_PHA_S[2:0] is used to modify the clock sampling edge. Details can refer to the following table.

Table 10-7 Output data start-edge Calibration (1x Clock Mode)

Clock Sampling Edge Tunning TX_CKOUT_PHA_S[2:0]	Clock Sampling Edge refer to Data Start-Edge (7x clock period)
000 (Default)	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

10.3.5 PLL Input Divider Value Setting

This is to set up the F_{PFD} is always around 5MHz (4~6MHz) so that the PLL loop bandwidth is almost constant at 100KHz. The input divider value range is 2 to 34, and it can be bypassed with INBYPASS = 1

Table 10-8 PLL Input Divider Configuration

Input Divider Setting	Input Divider Value (N_{IN})
INDIV[4]=0000000	2
INDIV[4]=0, INDIV[3:0]! \neq 0	$2 + \text{SUM}[(2^i) * \text{INDIV}[i]]$
INDIV[4]=1, INDIV[3:0]! \neq 0	$2 * (2 + \text{SUM}[(2^i) * \text{INDIV}[i]])$

$$F_{PFD} = F_{pixclk} / N_{IN}$$

10.3.6 PLL Feedback Divider Value Setting

The feedback divider value range is 8 to 260.

$F_{VCO} = (F_{IN} / N_{IN}) * N_{FBK}$, The F_{VCO} Frequency should be between 150MHz to 1GHz.

Table 10-9 PLL Feedback Divider Configuration

Feedback Divider Setting	Feedback Divider Value (N_{FBK})
PLLN[6:0]	$2 * \{ 2 + \text{Sum} [2^i * \text{PLLN}[i]] \}$ for $i = 0$ to 6 , PLLN[6:0] \neq 0 & 1
0000000	260
0000001	N/A
0000010	8
...	...
1111111	258

10.3.7 VCO Gain Calibration

The KVCO Setting value should be match the F_{VCO} Frequency

Table 10-10 PLL VCO gain calibration

VCO Frequency Band Setting (KVCO[1:0])	VCO Gain Calibration Value
00	150MHz to 400MHz VCO Frequency Range
01	400MHz to 650MHz VCO Frequency Range
10	650MHz to 900MHz VCO Frequency Range
11	900MHz to 1GHz VCO Frequency Range

10.3.8 PLL Output Clock Frequency Setting

Should make equals to pixel clock frequency multiply 7.

$$F_{OUT} = F_{VCO} / N_{OUT}$$

Table 10-11 PLL Post Divider Configuration

Post Divider Setting (POST_DIVA[1:0])	Post Divider Value (N_{OUT})
00	1
01	2
10	4 (Attention, not 3)
11	3 (Attention, not 4)

10.3.9 LVDS-LCDC RGB Data Mapping

LCDC	24bit	LVDS		LCDC	18bit	LVDS
------	-------	------	--	------	-------	------

VSYNC	-----	VSYNC	VSYNC	-----	VSYNC
HSYNC	-----	HSYNC	HSYNC	-----	HSYNC
DE	-----	DE	DE	-----	DE
R7	-----	R7	R7	-----	R7
R6	-----	R6	R6	-----	R6
R5	-----	R5	R5	-----	R5
R4	-----	R4	R4	-----	R4
R3	-----	R3	R3	-----	R3
R2	-----	R2	R2	-----	R2
R1	-----	R1	R1	x	R1
R0	-----	R0	R0	x	R0
G7	-----	G7	G7	-----	G7
G6	-----	G6	G6	-----	G6
G5	-----	G5	G5	-----	G5
G4	-----	G4	G4	-----	G4
G3	-----	G3	G3	-----	G3
G2	-----	G2	G2	-----	G2
G1	-----	G1	G1	x	G1
G0	-----	G0	G0	x	G0
B7	-----	B7	B7	-----	B7
B6	-----	B6	B6	-----	B6
B5	-----	B5	B5	-----	B5
B4	-----	B4	B4	-----	B4
B3	-----	B3	B3	-----	B3
B2	-----	B2	B2	-----	B2
B1	-----	B1	B1	x	B1
B0	-----	B0	B0	x	B0

10.3.10 LVDS VESA/JEIDA Mode

VESA Mode LVDS output timing like following figure.

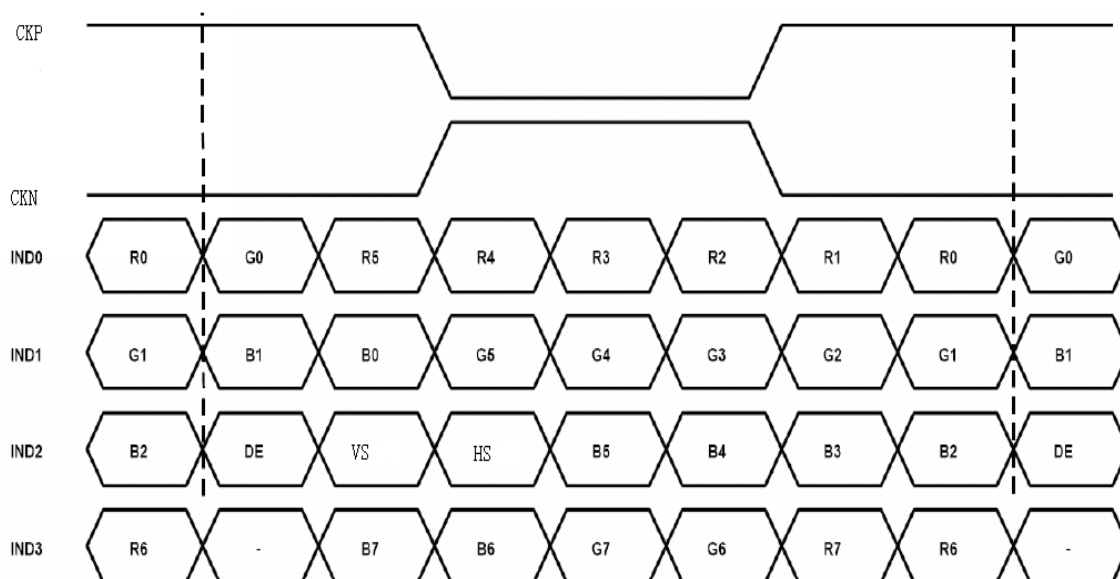


Figure 10-1 VESA Mode LVDS output timing

JEIDA Mode LVDS output timing like following figure.

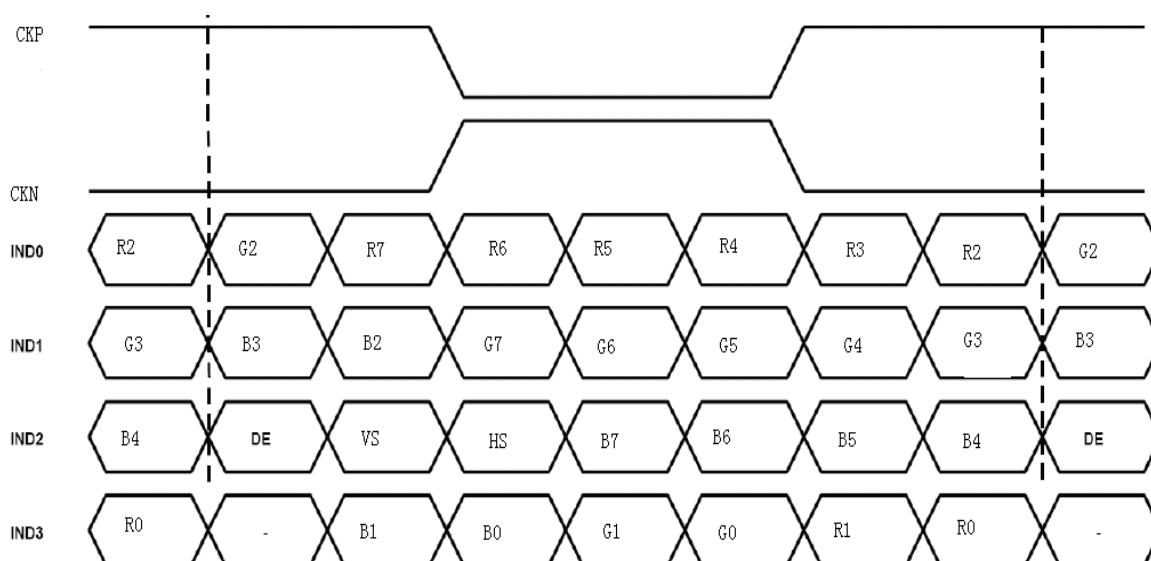


Figure 10-2 JEIDA Mode LVDS output timing

Note: If application set LCDC output 18bpp RGB data, the LVDS should set to JEIDA mode.

10.4 Operate Flow

10.4.1 Power Down Mode

When PLL_EN set to 0 and BG_PWD set to 0, The PLL will in low power consumption state, but the bandgap and reference generator is alive for fast wake up. When BG_PWD set to 1, The LVDS will enter the full power down mode for minimum power consumption

10.4.2 LVDS Output Mode

1. Follow used LVDS panel, Select LVDS output mode, VESA or JEIDA
2. Set PLL_EN and BG_PWD make LVDS leaves full power down mode.
3. Set TXCTRL follow 1.3.1
4. Set TXPLL0 and TXPLL1 follow 1.3.2 to 1.3.8
5. Wait LVDS_PLL_LOCK equal to 1
6. If LVDS panel can't display correctly, change TX_DLY_SEL value from 0 to 7, look the panel display status. If ok, the LVDS set is finish. If not, follow the step 7.
7. Set TX_CKOUT_PHA_S value from 0 to 7, and after each set, jump to step 6,change TX_DLY_SEL value from 0 to 7, until the LVDS panel display correctly.

10.4.3 RGB CMOS Output Mode

If you wants LVDS output RGB data at CMOS voltage, please Set the LVDS registers follow 10.3.1 table 10-3.

11 Camera Interface Module

11.1 Overview

The camera interface module (CIM) supports commonly available CMOS or CCD type image sensors. The CIM sources the digital image stream through a common 8-bit parallel digital protocol. The CIM can directly connect to external CMOS image sensors and ITU656 standard video decoders.

11.1.1 Features

- Input image size up to 4096x4096 pixels
- Max. VGA for image preview
- Max. VGA for video record
- Integrated DMA
- Supported data format: YCbCr 4:4:4, YCbCr 4:2:2 and other formats
- Output format: csc mode is YCbCr 4:2:2 or YCbCr 4:2:0, bypass mode is the input data format
- Output frame format
 - Packaged : for all data format
 - Separated: for YCbCr 4:4:4, YCbCr 4:2:2 and YCbCr 4:2:0
- Supports ITU656 (YCbCr 4:2:2) input
- Configurable CIM_VSYNC and CIM_HSYNC signals: active high/low
- Configurable CIM_PCLK: active edge rising/falling
- 256x33 image data receive FIFO (RXFIFO)
- PCLK max. 80MHz
- Configurable output order

11.1.2 Pin Description

Table 11-1 Camera Interface Pins Description

Name	I/O	Description
CIM_MCLK	O	CIM work clock
CIM_PCLK	I	Pixel clock from Image Sensor
CIM_VSYNC	I	Vertical synchronous from Image Sensor
CIM_HSYNC	I	Horizontal synchronous from Image Sensor
CIM_DATA[7:0]	I	Data bus from Image Sensor

11.2 CIM Special Register

11.2.1 CIM Register Map

The special registers are for CIM to configure and control the interface and DMA operation. The table below lists these registers.

Table 11-2 CIM Registers

Name	RW	Reset Value	Offset	Access Size
CIMCFG	RW	0x00000000	0x13060000	32
CIMCR	RW	0x00000000	0x13060004	32
CIMST	RW	0x00150002	0x13060008	32
CIMIID	R	0x00000000	0x1306000C	32
CIMDA	RW	0x00000000	0x13060020	32
CIMFA	R	0x00000000	0x13060024	32
CIMFID	R	0x00000000	0x13060028	32
CIMCMD	R	0x00000000	0x1306002C	32
CIMWSIZE	RW	0x00000000	0x13060030	32
CIMWOFFSET	RW	0x00000000	0x13060034	32
CIMYFA	R	0x00000000	0x13060038	32
CIMYCMD	R	0x00000000	0x1306003C	32
CIMCBFA	R	0x00000000	0x13060040	32
CIMBCMD	R	0x00000000	0x13060044	32
CIMCRFA	R	0x00000000	0x13060048	32
CIMRCMD	R	0x00000000	0x1306004C	32
CIMCR2	RW	0x00000000	0x13060050	32
CIMFS	RW	0x00000000	0x13060054	32
CIMIMR	RW	0x00000F0D	0x13060058	32
CIMTC	RW	0x00000000	0x1306005C	32
CIMTINX	RW	0x00000000	0x13060060	32
CIMTCNT	R	0x????????	0x13060064	32

11.2.2 CIM Configuration Register (CIMCFG)

CIMCFG																0x13060000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EOFEN	EXP	Reserved				BS	BW				SEP	ORDER		DF	INV_DAT	VSP	HSP	PCP	BURST_T	YPE	DUMMY	E_VSYNC	LM	PACK			FP	Reserved	DSM		
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW															
31	EEOFEN	Early EOF Mode Enable. 0: EEOF mode is disabled 1: When CIMCR.EEOF_LINE lines data has been transferred of a frame, the EEOF flag will be set, and the EEOF interrupt will occur	RW															
30	EXP	Expand mode for CIM_DATA width. 0: CIM_DATA width = 8 1: CIM_DATA width > 8																
29:24	Reserved	Writing has no effect. Read as zero.	R															
23:22	BW	Bus width of CIM_DATA Interface. When BW is n, the bus width is (n+9) bits.	RW															
20	SEP	Separate frame format enable. Used in output data format of YCbCr 4:4:4 and YcbCr 4:2:2. 0: Output is packaged frame format 1: Output is separated frame format	RW															
19:18	ORDER	Input data stream order. <table border="1" data-bbox="450 900 1035 1115"> <thead> <tr> <th></th> <th>YCbCr 4:4:4</th> <th>ITU656/YCbCr 4:2:2</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>YCbCr</td> <td>Y₀CbY₁Cr</td> </tr> <tr> <td>01</td> <td>YCrCb</td> <td>Y₀CrY₁Cb</td> </tr> <tr> <td>10</td> <td>CbCrY</td> <td>CbY₀CrY₁</td> </tr> <tr> <td>11</td> <td>CrCbY</td> <td>CrY₀CbY₁</td> </tr> </tbody> </table>		YCbCr 4:4:4	ITU656/YCbCr 4:2:2	00	YCbCr	Y ₀ CbY ₁ Cr	01	YCrCb	Y ₀ CrY ₁ Cb	10	CbCrY	CbY ₀ CrY ₁	11	CrCbY	CrY ₀ CbY ₁	RW
	YCbCr 4:4:4	ITU656/YCbCr 4:2:2																
00	YCbCr	Y ₀ CbY ₁ Cr																
01	YCrCb	Y ₀ CrY ₁ Cb																
10	CbCrY	CbY ₀ CrY ₁																
11	CrCbY	CrY ₀ CbY ₁																
17:16	DF	Input data format. 00: The data format has no effect. 01: YCbCr 4:4:4 10: YCbCr 4:2:2 11: ITU656 YCbCr 4:2:2	RW															
15	INV_DAT	Inverse every bit of input data. 0: not inverse; 1: inverse.	RW															
14	VSP	VSYNC polarity selection. When VSYNC signal is input from pin. CIM_VSYNC, this bit specifies the VSYNC signal active level and leading edge. When VSYNC is retrieved from SAV&EAV, this bit is ignored. 0: VSYNC signal active high, VSYNC signal leading edge is rising edge 1: VSYNC signal active low, VSYNC signal leading edge is falling edge	RW															
13	HSP	Specifies the HSYNC signal active level and leading edge. 0: HSYNC signal active high, HSYNC signal leading edge is rising edge 1: HSYNC signal active low, HSYNC signal leading edge is falling edge	RW															
12	PCP	Specifies the PCLK working edge. 0: Data is sampled by PCLK rising edge 1: Data is sampled by PCLK falling edge	RW															
11:10	BURST_ TYPE	DMA burst type. 00: INCR4 01: INCR8	RW															

		10: INCR16 11: INCR32																												
9	DUMMY	DUMMY zero function. When DUMMY is 1, CIM hardware adds one byte zero to every 3 input data bytes to form 32-bit data. 0: DUMMY zero function disabled 1: DUMMY zero function enabled	RW																											
8	E_VSYN C	External / internal VSYNC selection. When DSM is ITU656Progressive Mode, VSYNC can be external (provided by sensor) or internal (retrieved from SAV&EAV). This bit only valid for ITU656Progressive Mode; In other DSM modes, this bit should always be 0. 0: Internal VSYNC mode, pin CIM_VSYN is ignored 1: External VSYNC mode, VSYNC is provided by image sensor via pin CIM_VSYN	RW																											
7	LM	Line Mode for ITU656. 0: EAV is before SAV in each line 1: SAV is before EAV in each line	RW																											
6:4	PACK	Data packing mode, pack 8-bit input data into 32-bit data for FIFO. <table border="1" data-bbox="549 943 1211 1330"> <thead> <tr> <th>PACK</th> <th>Bypass Mode</th> <th>CSC Mode</th> </tr> </thead> <tbody> <tr> <td>3'b000</td> <td>0x 11 22 33 44</td> <td>0x Y₀ Cb Y₁ Cr</td> </tr> <tr> <td>3'b001</td> <td>0x 22 33 44 11</td> <td>0x Cb Y₁ Cr Y₀</td> </tr> <tr> <td>3'b010</td> <td>0x 33 44 11 22</td> <td>0x Y₁ Cr Y₀Cb</td> </tr> <tr> <td>3'b011</td> <td>0x 44 11 22 33</td> <td>0x Cr Y₀ Cb Y₁</td> </tr> <tr> <td>3'b100</td> <td>0x 44 33 22 11</td> <td>0x Cr Y₁ Cb Y₀</td> </tr> <tr> <td>3'b101</td> <td>0x 33 22 11 44</td> <td>0x Y₁ Cb Y₀ Cr</td> </tr> <tr> <td>3'b110</td> <td>0x 22 11 44 33</td> <td>0x Cb Y₀ Cr Y₁</td> </tr> <tr> <td>3'b111</td> <td>0x 11 44 33 22</td> <td>0x Y₀ Cr Y₁ Cb</td> </tr> </tbody> </table> <p>In this table, 0x11, 0x22, 0x33 and 0x44 mean the received data from the sensor, 0x11 is received first and 0x44 is received last, and Y₀ is received before Y₁.</p>	PACK	Bypass Mode	CSC Mode	3'b000	0x 11 22 33 44	0x Y ₀ Cb Y ₁ Cr	3'b001	0x 22 33 44 11	0x Cb Y ₁ Cr Y ₀	3'b010	0x 33 44 11 22	0x Y ₁ Cr Y ₀ Cb	3'b011	0x 44 11 22 33	0x Cr Y ₀ Cb Y ₁	3'b100	0x 44 33 22 11	0x Cr Y ₁ Cb Y ₀	3'b101	0x 33 22 11 44	0x Y ₁ Cb Y ₀ Cr	3'b110	0x 22 11 44 33	0x Cb Y ₀ Cr Y ₁	3'b111	0x 11 44 33 22	0x Y ₀ Cr Y ₁ Cb	6:4
PACK	Bypass Mode	CSC Mode																												
3'b000	0x 11 22 33 44	0x Y ₀ Cb Y ₁ Cr																												
3'b001	0x 22 33 44 11	0x Cb Y ₁ Cr Y ₀																												
3'b010	0x 33 44 11 22	0x Y ₁ Cr Y ₀ Cb																												
3'b011	0x 44 11 22 33	0x Cr Y ₀ Cb Y ₁																												
3'b100	0x 44 33 22 11	0x Cr Y ₁ Cb Y ₀																												
3'b101	0x 33 22 11 44	0x Y ₁ Cb Y ₀ Cr																												
3'b110	0x 22 11 44 33	0x Cb Y ₀ Cr Y ₁																												
3'b111	0x 11 44 33 22	0x Y ₀ Cr Y ₁ Cb																												
3	FP	Field flag polarity selection. When ITU656 progressive stream is input, this bit specifies the field flag active level. When other modes are used, this bit is ignored. 0: Field flag active low 1: Field flag active high	RW																											
2	Reserved	Writing has no effect. Read as zero.	R																											
1:0	DSM	Data sample mode. Please refer to the table below. <table border="1" data-bbox="612 1778 1144 1986"> <thead> <tr> <th>DSM</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>2'b00</td> <td>ITU656Progressive Mode</td> </tr> <tr> <td>2'b01</td> <td>ITU656Interlace Mode</td> </tr> <tr> <td>2'b10</td> <td>Gated Clock Mode</td> </tr> <tr> <td>2'b11</td> <td>Reserved</td> </tr> </tbody> </table>	DSM	Description	2'b00	ITU656Progressive Mode	2'b01	ITU656Interlace Mode	2'b10	Gated Clock Mode	2'b11	Reserved	RW																	
DSM	Description																													
2'b00	ITU656Progressive Mode																													
2'b01	ITU656Interlace Mode																													
2'b10	Gated Clock Mode																													
2'b11	Reserved																													

11.2.3 CIM Control Register (CIMCR)

CIMCR																0x13060004																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	EEOF_LINE												FRC				Reserved	WINE	Reserved						DMA_SYNC	MBEN	H_SYNC	Reserved	SW_RST	DMA_EN	RF_RST	ENA	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:20	EEOF_LINE	When EEOF_LINE lines data has been transferred of a frame, the EEOF flag will be set, and the EEOF interrupt will occur.	RW
19:16	FRC	CIM frame rate control. If FRC = n, CIM sampling one frame from every (n+1) frames from the sensor.	RW
15	Reserved	Read as zero.	R
14	WINE	To enable window-image. Used to indicate whether the registers CIMWSIZE and CIMWOFFSET work or not. 0: the value in CIMWSIZE and CIMWOFFSET will be ignored 1: the value in CIMWSIZE and CIMWOFFSET will be used	RW
13:8	Reserved	Read as zero.	R
7	DMA_SYNC	The control bit to enable DMA synchronization. 0: The valid data input to CIM will be transferred by DMA to external memory 1: When a new descriptor-DMA transfer starts with writing CIMDA, a frame synchronization will be done, and the data in RXFIFO will be ignored	RW
6	MBEN	Macro Block Mode Enable. It effects the output data storage only when the output data format is YUV420. 0: The output data of a frame is separated to Y, U and V. 1: The data storage is according to the 16x16 macro blocks.	RW
5	H_SYNC	Horizontal Sync Enable. 0: disable 1: enable It is only used when CIMCFG.SEP is 1.	RW
4	Reserved	Writing has no effect, read as zero.	R
3	SW_RST	Software reset enable. 0: Don't care 1: Reset the CIM module.	RW
2	DMA_EN	Enable / disable the DMA function. 0: disable DMA; 1: enable DMA.	RW

1	RF_RST	RXFIFO software reset. Setting 1 to RXF_RST can reset RXFIFO immediately. Setting 0 to RXF_RST can stop resetting RXFIFO. After reset, RXFIFO is empty.	RW
0	ENA	Enable or disable the CIM module. 0: CIM is not enabled, or disable CIM immediately 1: CIM is enabled, or enabling CIM	RW

11.2.4 CIM Control Register 2 (CIMCR2)

CIMCR2																0x13060050																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved						FRAGHE	FRAGVE	FSC	ARIF	HRS	VRS	CSC							OP	Reserved	OPE	EMM	APM								
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:26	Reserved	Writing has no effect, read as zero.	R
25	FRAGHE	When horizontal frame scale is enabled and the frame horizontal size is not the multiple of the down scale, this bit decides the remainder will be ignored or not. 0: not ignore 1: ignore	RW
24	FRAGVE	When vertical frame scale is enabled and the frame vertical size is not the multiple of the down scale, this bit decides the remainder will be ignored or not. 0: not ignore 1: ignore	RW
23	FSC	Enable frame size check. 0: Disable frame size check. 1: Enable frame size check.	RW
22	ARIF	Enable auto recovery for incomplete frame. This field will effect only when FSC is 1. 0: Disable auto recovery. 1: Enable auto recovery.	RW
21:20	HRS	Specified the horizontal ratio for down scale. 00: no scale 01: 1/2 down scale 10: 1/4 down scale 11: 1/8 down scale	RW

19:18	VRS	Specified the vertical ratio for down scale. 00: no scale 01: 1/2 down scale 10: 1/4 down scale 11: 1/8 down scale	RW																																							
17:16	CSC	CSC Mode Select. 00: Bypass mode. 01: Reserved. 10: CSC to YCbCr 422. 11: CSC to YCbCr 420.	RW																																							
15:6	Reserved	Writing has no effect, read as zero.	R																																							
5:4	OP	Optional Priority Configuration. Only used when OPE is set to 1. <table border="1" data-bbox="523 763 1315 1518"> <thead> <tr> <th>PG</th> <th>CIM AHB Priority</th> <th>Number of Data in FIFO</th> </tr> </thead> <tbody> <tr> <td rowspan="4">2'b00</td> <td>0</td> <td>$n \leq 8$</td> </tr> <tr> <td>1</td> <td>$8 < n \leq 16$</td> </tr> <tr> <td>2</td> <td>$16 < n \leq 32$</td> </tr> <tr> <td>3</td> <td>$32 < n$</td> </tr> <tr> <td rowspan="4">2'b01</td> <td>0</td> <td>$n \leq 16$</td> </tr> <tr> <td>1</td> <td>$16 < n \leq 32$</td> </tr> <tr> <td>2</td> <td>$32 < n \leq 64$</td> </tr> <tr> <td>3</td> <td>$64 < n$</td> </tr> <tr> <td rowspan="4">2'b10</td> <td>0</td> <td>$n \leq 32$</td> </tr> <tr> <td>1</td> <td>$32 < n \leq 64$</td> </tr> <tr> <td>2</td> <td>$64 < n \leq 96$</td> </tr> <tr> <td>3</td> <td>$96 < n$</td> </tr> <tr> <td rowspan="4">2'b11</td> <td>0</td> <td>$n \leq 64$</td> </tr> <tr> <td>1</td> <td>$64 < n \leq 96$</td> </tr> <tr> <td>2</td> <td>$96 < n \leq 128$</td> </tr> <tr> <td>3</td> <td>$128 < n$</td> </tr> </tbody> </table> It is suggested to use 2'b10.	PG	CIM AHB Priority	Number of Data in FIFO	2'b00	0	$n \leq 8$	1	$8 < n \leq 16$	2	$16 < n \leq 32$	3	$32 < n$	2'b01	0	$n \leq 16$	1	$16 < n \leq 32$	2	$32 < n \leq 64$	3	$64 < n$	2'b10	0	$n \leq 32$	1	$32 < n \leq 64$	2	$64 < n \leq 96$	3	$96 < n$	2'b11	0	$n \leq 64$	1	$64 < n \leq 96$	2	$96 < n \leq 128$	3	$128 < n$	RW
PG	CIM AHB Priority	Number of Data in FIFO																																								
2'b00	0	$n \leq 8$																																								
	1	$8 < n \leq 16$																																								
	2	$16 < n \leq 32$																																								
	3	$32 < n$																																								
2'b01	0	$n \leq 16$																																								
	1	$16 < n \leq 32$																																								
	2	$32 < n \leq 64$																																								
	3	$64 < n$																																								
2'b10	0	$n \leq 32$																																								
	1	$32 < n \leq 64$																																								
	2	$64 < n \leq 96$																																								
	3	$96 < n$																																								
2'b11	0	$n \leq 64$																																								
	1	$64 < n \leq 96$																																								
	2	$96 < n \leq 128$																																								
	3	$128 < n$																																								
3	Reserved	Writing has no effect, read as zero.	R																																							
2	OPE	Optional Priority Mode Enable Control. Only used when APM is 1. 0: CIM calculates the priority according to the fifo status 1: CIM calculates the priority according to OPG which is configured by software	RW																																							
1	EME	Emergency Mode Enable Control. 0: Emergency Mode Disable 1: Emergency Mode Enable	RW																																							
0	APM	Auto Priority Mode Enable Control. 0: Auto priority mode disable. CIM uses the priority set by arbiter	RW																																							

		1: Auto priority mode enable. CIM can use the priority according the fifo status	
--	--	--	--

11.2.5 CIM Status Register (CIMST)

CIMST																0x13060008																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	Reserved										Cr_RFOF	Cr_RFE	Cb_RFOF	Cb_RFE	Y_RFOF	Y_RFE	Reserved					DEEOF	DSTP	DEOF	DSOF	Reserved			TLBE	FSE	RFOF	RFE	VDD	
RST	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	Name	Description	RW
31:22	Reserved	Writing has no effect, read as zero.	R
21	Cr_RFOF	Cr_RXFIFO over flow. When Cr_RXFIFO over flow happens, Cr_RX_OF is set 1. Can generate interrupt if CIMIMR.RFOFM bit is set. Write 0 to this bit to clear.	RW
20	Cr_RFE	Cr_RXFIFO empty. Indicates whether Cr_RXFIFO is empty or not. After reset, RXFIFO is empty, and Cr_RX_EMPTY is 1. 0: Cr_RXFIFO is not empty 1: Cr_RXFIFO is empty	R
19	Cb_RFOF	Cb_RXFIFO over flow. When Cb_RXFIFO over flow happens, Cb_RX_OF is set 1. Can generate interrupt if CIMIMR.RFOFM bit is set. Write 0 to this bit to clear.	RW
18	Cb_RFE	Cb_RXFIFO empty. Indicates whether Cb_RXFIFO is empty or not. After reset, Cb_RXFIFO is empty, and Cb_RX_EMPTY is 1. 0: Cb_RXFIFO is not empty 1: Cb_RXFIFO is empty	R
17	Y_RFOF	Y_RXFIFO over flow. When Y_RXFIFO over flow happens, Y_RX_OF is set 1. Can generate interrupt if CIMIMR.RFOFM bit is set. Write 0 to this bit to clear.	RW
16	Y_RFE	Y_RXFIFO empty. Indicates whether Y_RXFIFO is empty or not. After reset, Y_RXFIFO is empty, and Y_RX_EMPTY is 1. 0: Y_RXFIFO is not empty 1: Y_RXFIFO is empty	R
15:12	Reserved	Writing has no effect, read as zero.	R

11	DDEOF	When set to 1, indicates the DMA has transferred CIMCTRL.EEOF_LINE lines data of a frame. Can generate an interrupt if CIMIMR.DDEOFM is 0. Write 0 to clear.	RW
10	DSTOP	When set to 1, indicates the DMA complete transferring data and stop the operation. Can generate an interrupt if CIMIMR.DSTPM is 0. Write 0 to clear.	RW
9	DEOF	When set to 1, indicates the DMA complete a transfer from RXFIFO to a frame buffer. Can generate an interrupt if CIMIMR. DEOFM is 0. Write 0 to clear.	RW
8	DSOF	When set to 1, Indicates the DMA start a transfer from RXFIFO to a frame buffer. Can generate an interrupt if CIMIMR. DSOFM is 0. Write 0 to clear.	RW
7:5	Reserved	Read as zeros.	R
4	TLBE	TLB error. When set to 1, Indicates there is a TLB error. Write 0 to clear.	RW
3	FSE	Indicates that frame size check error. 0: No frame size check error occurs. 1: The received size of frame is not equal to the expected. Can generate an interrupt if CIMIMR.FSEM is 0. Write 0 to clear.	RW
2	RFOF	Indicates RXFIFO overflow. 0: RXFIFO is not overflow. 1: RXFIFO is overflow. Can generate an interrupt if CIMIMR.ROFM is 0. Write 0 to clear.	RW
1	RFE	Indicates RXFIFO is empty. 0: RXFIFO is not empty 1: RXFIFO is empty	R
0	VDD	CIM disable done. Indicate this module is disabled after clear the CIMCR.ENA bit to disable the CIM module. 0: CIM has not been disabled. 1: CIM has been disabled. Can generate an interrupt if CIMIMR.VDDM is 0. Write 0 to this bit to clear.	RW

11.2.6 CIM Interrupt Mask Register (CIMIMR)

CIMIMR																0x13060058																									
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
	Reserved																																								
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	1									

Bits	Name	Description	RW
31:12	Reserved	Read as zeros.	R
11	DEEOFM	The control bit to mask DMA EEOF interrupt. 0: enable 1: mask	RW
10	DSTPM	The control bit to mask DMA STOP interrupt. 0: enable 1: mask	RW
9	DEOFM	The control bit to mask DMA EOF interrupt. 0: enable 1: mask	RW
8	DSOFM	The control bit to mask DMA SOF interrupt. 0: enable 1: mask	RW
7:5	Reserved	Read as zeros.	R
4	TLBEM	This bit to control TLB error interrupt. 0: enable 1: mask	RW
3	FSEM	The control bit to mask frame size check error interrupt. 0: enable 1: mask	RW
2	RFOFM	The control bit to mask RXFIFO overflow interrupt. 0: enable 1: mask	RW
1	Reserved	Read as zeros.	R
0	VDDM	The control bit to mask VDD interrupt. 0: enable 1: mask	RW

11.2.7 CIM Interrupt ID Register (CIMIID)

CIMIID																0x1306000C																					
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
	FID																																				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	FID	Interrupt frame ID. Contains a copy of the Frame ID register (CIMFID) from the descriptor currently being processed when a DMA_SOF or DMA_EOF interrupt is generated. CIMIID is written to only when CIMCMD.SOFINT or CIMCMD.EOFINT is high. As such, the register is considered to be sticky and will be overwritten only when the associated interrupt is cleared by writing the CIM state register.	R

11.2.8 CIM Descriptor Address (CIMDA)

CIMIDA																0x13060020																						
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
	NDA																																					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	NDA	Next descriptor physical address in external memory. DMAC gets the next descriptor according to it after finishing the current one. The target address Bits [3:0] must be zero to be aligned to 16-byte boundary.	RW

11.2.9 CIM Frame buffer Address Register (CIMFA)

CIMIFA																0x13060024																							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
	FA																																						
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	FA	Frame buffer virtual address in external memory when CIMCFG. SEP is 0. When starts CIM, DMA transfers data from RXFIFO to frame buffer. This address is increased by hardware automatically. Bits [6:0] must be zero to be aligned to 32-word boundary.	R

NOTE: CIMFA comes from DMA Descriptor, so here it is read-only.

11.2.10 CIM Frame ID Register (CIMFID)

CIMFID																0x13060028																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	FID																																
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	FID	Frame ID. The particular use of this field is up to the software. This ID will be copied to the CIMIID register when an interrupt occurs.	R

NOTE: CIMFID comes from DMA Descriptor, so here it is read-only.

11.2.11 CIM DMA Command Register (CIMCMD)

CIMFID																0x1306002C																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	SOFINTEn	EOFINTEn	EOFINTEn	STOP	Reserved					LEN																							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW
31	SOFINTEn	Interrupt enable for DMA starting a frame-buffer transfer. 1: DMA will set CIMSTATE.DMA_SOF when start of a frame-buffer transfer When one frame uses several buffers, it is suggested to set SOFINTEn of first buffer only.	R
30	EOFINTEn	Interrupt enable for DMA ending a frame-buffer transfer.	R

		1: DMA will set CIMSTATE.DMA_EOF when CIMCMD.LEN is decreased to 0, which means end of a frame-buffer transfer When one frame uses several buffers, it is suggested to set EOFINTEn of last buffer only.	
29	EEOFINTEn	Interrupt enable for DMA issuing an earlier eof interrupt.	R
28	STOP	DMA stop. When DMA complete transferring data, STOP bit decides whether DMA should loading next descriptor or not. 0: DMA start loading next descriptor 1: DMA stopped, and CIMSTATE.DMA_STOP bit is set 1 by hardware	R
27	OFRCVEN	Auto recovery enable when there is RXFIFO overflow. 0: No auto recovery when overflow occurs, thus the software should do something 1: Auto recovery enable, the hardware will correct the overflow	R
26:24	Reserved	Writing has no effect, read as zero.	R
23:0	LEN	Length of transfer in words. Indicate the number of words to be transferred by DMA to a frame buffer. LEN = 0 is not valid. DMA transfers data according to LEN. Each time one or more word(s) been transferred, LEN is decreased automatically.	R

11.2.12 CIM Window Size (CIMWSIZE)

	CIMWSIZE																0x13060030															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				LPF												Reserved				PPL											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:29	Reserved	Writing has no effect, read as zero.	R
28:16	LPF	Lines per frame for CIM output.	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	PPL	PPL must be multiples of 2. In fact, the number of CIM output data in word is equal to PPL/2.	RW

NOTE:

- When CIMCFG.SEP is 1, the total pixel number of window-size must be multiple of 4 or 8.
- When output data format is YCBCR4:4:4, it must be multiple of 4.
 - When output data format is YCBCR4:2:2, it must be multiple of 8.

11.2.13 CIM Window Offset (CIMWOFFSET)

CIMWOFFSET																0x13060034																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			V_OFFSET												Reserved			H_OFFSET													
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:29	Reserved	Writing has no effect, read as zero.	R
28:16	V_OFFSET	Vertical offset.	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	H_OFFSET	Horizontal offset. It should be an even number.	RW

11.2.14 CIM Frame Size Register (CIMFS)

This register will indicate how many pixels and bytes are included in a frame. The maximum frame size is 8192*8192.

CIMFS																0x13060054																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			FVS												BPP		Reserved			FHS											
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:29	Reserved	Writing has no effect, read as zero.	R
28:16	FVS	Vertical size of the frame N indicates the vertical includes (n+1) pixels.	RW
15:14	BPP	Number of bytes per pixel. 00: One pixel includes 1 byte. 01: One pixel includes 2 bytes. 10: One pixel includes 3 bytes. 11: One pixel includes 4 bytes.	RW
13	Reserved	Writing has no effect, read as zero.	R
12:0	FHS	Horizontal size of the frame. N indicates the horizontal includes (n+1) pixels.	RW

11.2.15 CIM Y Frame buffer Address Register (CIMYFA)

CIMYFA																0x13060038																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
YFA																																	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW
31:0	YFA	Y Frame buffer virtual address in external memory when CIMCFG. SEP is 1. When starts CIM, DMA transfers data from Y_RXFIFO to frame buffer. This address is increased by hardware automatically. Bits [7:0] must be zero to be aligned to 32-word boundary when the data format is YUV420; Otherwise, bits [6:0] must be zero to be aligned to 32-word boundary.	R

11.2.16 CIM Y DMA Command Register (CIMYCMD)

CIMYCMD																0x1306003C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SOFINT	EOFINT	EEOFINT _{En}	STOP	OFRCVEN	Reserved										YLEN																
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	SOFINT _{En}	Interrupt enable for DMA starting a frame-buffer transfer. 1: DMA will set CIMSTATE.DMA_SOF when start of a frame-buffer transfer When one frame uses several buffers, it is suggested to set SOFINT _{En} of first buffer only.	R
30	EOFINT _{En}	Interrupt enable for DMA ending a frame-buffer transfer. 1: DMA will set CIMSTATE.DMA_EOF when CIMYCMD.YLEN and CIMCbCMD.CbLEN and CIMCrCMD.CrLEN are decreased to 0, which means end of a frame-buffer transfer When one frame uses several buffers, it is suggested to set EOFINT _{En} of last buffer only.	R
29	EEOFINT _{En}	Interrupt enable for DMA issuing an earlier eof interrupt.	R
28	STOP	DMA stop. When DMA complete transferring data, STOP bit decides	R

		whether DMA should loading next descriptor or not. 0: DMA start loading next descriptor 1: DMA stopped, and CIMSTATE.DMA_STOP bit is set 1 by hardware	
27	OFRCVEN	Auto recovery enable when there is RXFIFO overflow. 0: No auto recovery when overflow occurs, thus the software should do something 1: Auto recovery enable, the hardware will correct the overflow DMA will do a frame synchronization, and retransfer the current descriptor.	
26:24	Reserved	Writing has no effect, read as zero.	R
23:0	YLEN	Length of transfer in words. Indicate the number of words to be transferred by DMA to a frame buffer. YLEN = 0 is not valid. DMA transfers data according to YLEN. Each time one or more word(s) been transferred, YLEN is decreased automatically.	R

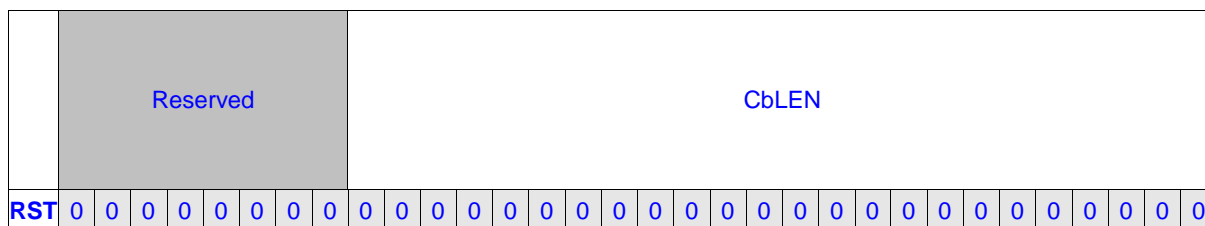
11.2.17 CIM Cb Frame buffer Address Register (CIMCBFA)

CIMCBFA																0x13060040																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	CbFA																																	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	CbFA	Cb Frame buffer virtual address in external memory when CIMCFG. SEP is 1. When starts CIM, DMA transfers data from Cb_RXFIFO to frame buffer. This address is increased by hardware automatically. Bits [6:0] must be zero to be aligned to 32-word boundary.	R

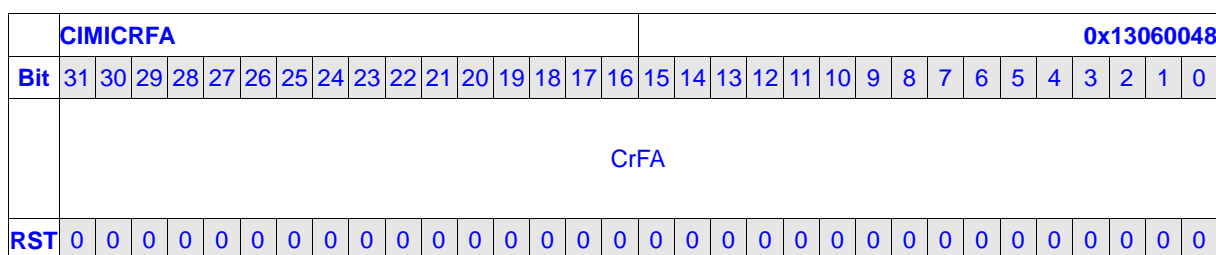
11.2.18 CIM Cb DMA Command Register (CIMCBCMD)

CIMCBCMD																0x13060044																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



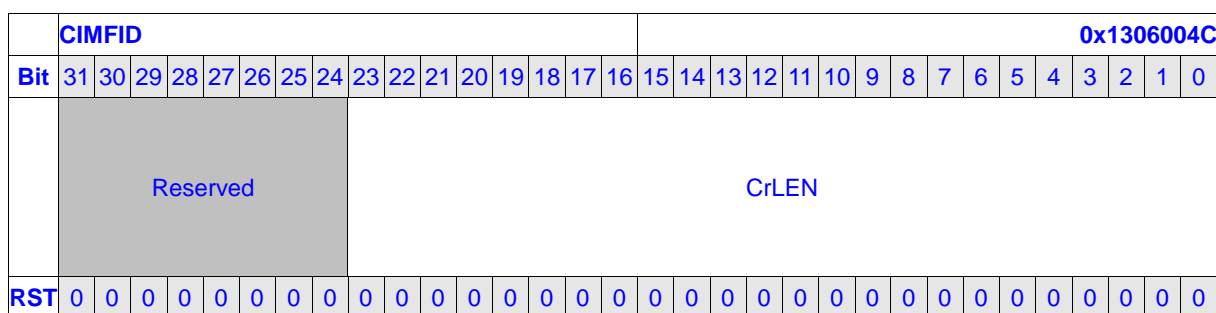
Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:0	CbLEN	Cb Length of transfer in words. Indicate the number of words to be transferred by DMA to a frame buffer. CbLEN = 0 is not valid. DMA transfers data according to CbLEN. Each time one or more word(s) been transferred, CbLEN is decreased automatically.	R

11.2.19 CIM Cr Frame buffer Address Register (CIMCRFA)



Bits	Name	Description	RW
31:0	CrFA	Cr Frame buffer virtual address in external memory when CIMCFG. SEP is 1. When starts CIM, DMA transfers data from Cr RXFIFO to frame buffer. This address is increased by hardware automatically. CrFA should be (CIMCBFA. CbFA + 64) when the data format is YUV420; Otherwise, bits [6:0] must be zero to be aligned to 32-word boundary.	R

11.2.20 CIM Cr DMA Command Register (CIMCRCMD)



Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:0	CrLEN	Cr Length of transfer in words. Indicate the number of words to be transferred by DMA to a frame buffer. CrLEN = 0 is not valid. DMA transfers data according to CrLEN. Each time one or more word(s) been transferred, CrLEN is decreased automatically.	R

11.2.21 CIM TLB Control Register (CIMTC)

CIMTC																0x1306005C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBA																RST	ENB														
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:2	TBA	TLB entry physical address. It should be word aligned, and TBA includes from bit 31 to bit2 only, the low two bits are zeros.	RW
1	RST	TLB reset. 0: Do nothing. 1: Reset TLB logic.	RW
0	ENB	Enable TLB or not. 0: TLB is disabled. 1: TLB is enabled and valid.	RW

11.2.22 CIM TLB Index Register (CIMTINX)

CIMTINX																0x13060060																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																Index															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Read as zeros.	R
2:0	Index	It will be used with CIMCNT together. This field will be used to	RW

		indicating which buffer will be needed. 0: Indicates VA of entry buffer 0 1: Indicates VA of entry buffer 1 2: Indicates VA of entry buffer 2 3: Indicates VA of entry buffer 3 4: Indicates PA of entry buffer 0 5: Indicates PA of entry buffer 1 6: Indicates PA of entry buffer 2 7: Indicates PA of entry buffer 3	
--	--	---	--

Note:

This register only is used for debug.

11.2.23 CIM TLB Content Register (CIMTCNT)

	CIMTCNT																0x13060064																															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
	CNT																																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	CNT	Read the register will get the content corresponding to CIMTINX.Index	R

11.3 CIM Data Sampling Modes

CIM module supports several types of data sampling mode. The modes and the corresponding signals used are shown in the following diagram:

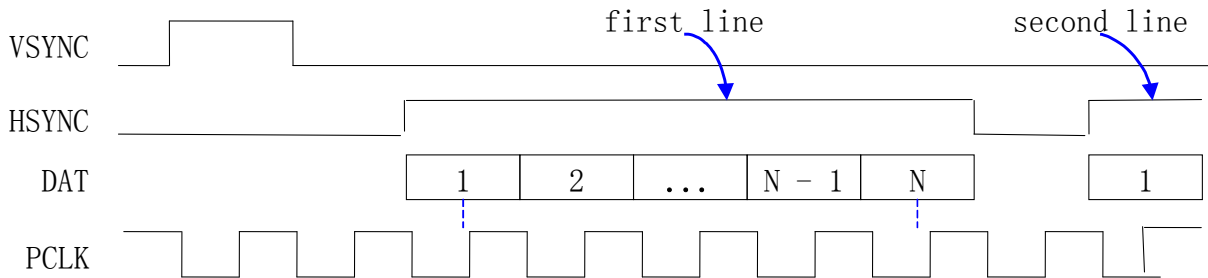
Table 11-3 The modes and the corresponding signals used

Mode \ Signals	CIM_VSYNC	CIM_HSYNC	CIM_PCLK	CIM_DATA
Gated Clock Mode	Y	Y	Y	Y
ITU656 Interlace Mode	N	N	Y	Y
ITU656 Progressive Mode	N	N	Y	Y

11.3.1 Gated Clock Mode

CIM_VSYNC, CIM_HSYNC, and CIM_PCLK signals are used in this mode.

A frame starts with VSYNC leading edge, then HSYNC goes active and holds the entire line. Data is sampled at the valid edge of PCLK when HSYNC is active; That means, HSYNC functions like “data enable” signal. Please refer to the figure below.



Gated Clock Mode Input Timing Diagram

The VSYNC leading edge, HSYNC active HIGH or LOW, PCLK valid edges are programmable.

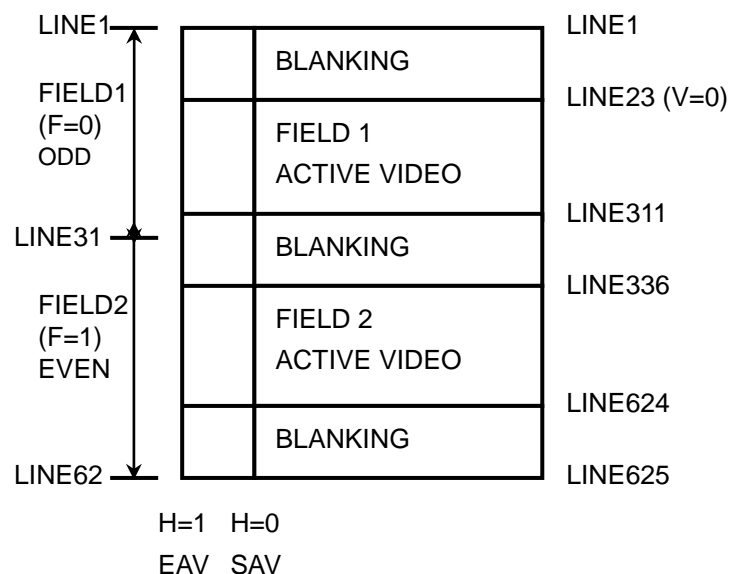
11.3.2 ITU656 Interlace Mode

In this mode, CIM_PCLK and CIM_DAT signals are used, CIM_VSYNC, CIM_HSYNC signals are ignored.

CIM utilizes the SAV & EAV code within ITU656data stream to get active video data.

The following diagrams and tables are quoted from ITU656standard. For more information about ITU656, please refer to ITU656 standard.

11.3.2.1 PAL Timing



LINE NUMBER	F	V	H (EAV)	H (SAV)	P0, P1, P2, P3
1-22	0 Field 1	1: blanking	1: in EAV, to indicate the end of active video	0: in SAV, to indicate the start of active video	Protection bits
23-310		0: video data			
311-312		1: blanking			
313-335	1 Field 2	1: blanking			
336-623		0: video data			
624-625		1: blanking			

Figure 11-1 Typical BT.656 Vertical Blanking Intervals for 625/50 Video Systems

11.3.2.2 Coding for Protection Bits

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

11.3.3 ITU656 Progressive Mode

CIM_PCLK and CIM_DAT signals are used in this mode. CIM_HSYNC signal is ignored.

CIM_VSYNC is optional in this mode. When the start of frame information is retrieved from SAV and EAV, it is known as internal VSYNC mode. When CIM_VSYNC is provided by sensor directly, it is known as external VSYNC mode. CIM supports both internal and external VSYNC modes.

ITU656Progressive Mode is a kind of Non-Interlace Mode. The image data are encoded within only one field. Most sensors support ITU656Progressive Mode.

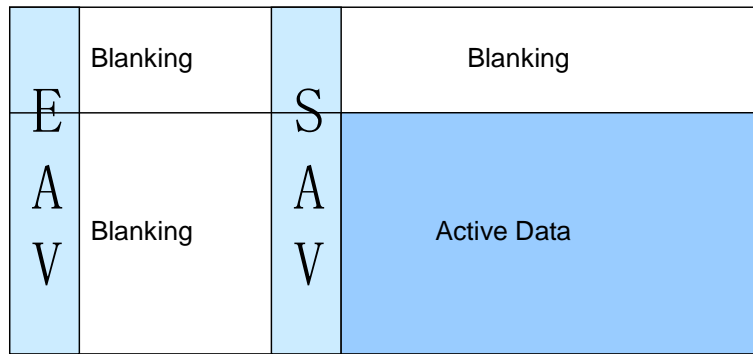


Figure 11-2 ITU656 Progressive Mode

11.4 DMA Descriptors

11.4.1 4-Word Descriptor

Used when output is packaged frame format.

A DMA descriptor is a 4-word block corresponding to the four DMA registers – CIMDA, CIMFA, CIMFID, and CIMCMD, aligned on 4-word (16-byte) boundary, in external memory:

- word [0] contains the physical address for next CIMDA
- word [1] contains the value for CIMFID
- word [2] contains the physical address for CIMFA
- word [3] contains the value for CIMCMD

Software must write the physical address of the first descriptor to CIMDA before enabling the CIM. Once the CIM is enabled, the first descriptor is read, and all 4 registers are written by the DMAC. The next DMA descriptor pointed to by CIMDA is loaded into the registers after all data for the current descriptor has been transferred.

11.4.2 8-Word Descriptor

Used when output is separated frame format.

A DMA descriptor is a 8-word block corresponding to the four DMA registers – CIMDA, CIMFA, CIMFID, and CIMCMD, aligned on 8-word (32-byte) boundary, in external memory:

- word [0] contains the physical address for next CIMDA
- word [1] contains the value for CIMFID
- word [2] contains the physical address for CIMYFA
- word [3] contains the value for CIMYCMD
- word [4] contains the physical address for CIMCBFA

- word [5] contains the value for CIMCBCMD
- word [6] contains the physical address for CIMCRFA
- word [7] contains the value for CIMCRCMD

Software must write the physical address of the first descriptor to CIMDA before enabling the CIM. Once the CIM is enabled, the first descriptor is read, and all 8 registers are written by the DMAC. The next DMA descriptor pointed to by CIMDA is loaded into the registers after all data for the current descriptor has been transferred.

NOTE: If only one frame buffer is used in external memory, the CIMDA field (word [0] of the DMA descriptor) must point back to itself. That is to say, the value of CIMDA is the physical address of itself.

11.5 Interrupt Generation

CIM has next interrupt sources:

Step 1. RXFIFO Over Flow Interrupt. (RFOF)

When the valid data number of RXFIFO reaches 32 and one more data are written to RXFIFO, CIMST.RFOF bit is set. At the same time, if RFOFM is 1, RFOF interrupt is generated.

Step 2. DMA Start Of Frame Data Transferring Interrupt. (DMA_SOF)

When the CIMCMD.SOFINT bit is 1 and DMA start transferring the first data from RXFIFO to frame buffer, CIMST.DMA_SOF bit is set. At the same time, if DMA_SOFM is 1, DMA_SOF interrupt is generated.

Step 3. DMA End Of Frame Data Transferring Interrupt. (DMA_EOF)

When the CIMCMD.EOFINT bit is 1 and DMA complete transferring the last data from RXFIFO to frame buffer, CIMST.DMA_EOF bit is set. At the same time, if DMA_EOFM is 1, DMA_EOF interrupt is generated.

Step 4. DMA Stop Transferring Interrupt. (DMA_STOP)

When the CIMCMD.STOP bit is 1 and DMA complete transferring the last data from RXFIFO to frame buffer, CIMST.DMA_STOP bit is set. At the same time, if DMA_STOPM is 1, DMA_STOP interrupt is generated.

Step 5. CIM Disable Done Interrupt. (VDD)

When disable the module by clearing the CIMCR.ENA, the module should be disabled after transferring current valid data. Then set the CIMST.VDD bit, at the same time, if VDDM is set, VDD interrupt is generated.

11.6 Software Operation

11.6.1 Enable CIM with DMA

- Step 1. Configure register CIMCFG.
- Step 2. Prepare frame buffer and descriptors.
- Step 3. Configure register CIMDA.
- Step 4. Clear state register: write 0 to register CIMSTATE.
- Step 5. Reset RXFIFO: configure register CIMCTRL with DMA_EN=1, RXF_RST=1, ENA=0.
- Step 6. Stop resetting RXFIFO: configure register CIMCTRL with DMA_EN=1, RXF_RST=0, ENA=0.
- Step 7. Enable CIM: configure register CIMCTRL with DMA_EN=1, RXF_RST=0, ENA=1.

11.6.2 Enable CIM without DMA

- 1 Configure register CIMCFG.
- 2 Clear state register: write 0 to register CIMSTATE.
- 3 Reset RXFIFO: configure register CIMCTRL with DMA_EN=0, RXF_RST=1, ENA=0.
- 4 Stop resetting RXFIFO: configure register CIMCTRL with DMA_EN=0, RXF_RST=0, ENA=0.
- 5 Enable CIM: configure register CIMCTRL with DMA_EN=0, RXF_RST=0, ENA=1.

11.6.3 Disable CIM

Method 1:

- Step 1. Configure register CIMCTRL with RXF_RST=0, ENA=0. // quick disable
- Step 2. Clear state register: write 0 to register CIMSTATE.

Method 2:

When DMA is enabled, the following sequence is recommended:

- Step 1. Configure descriptor with STOP = 1.
- Step 2. Wait DMA_STOP interrupt, then write 0 to CIMCTRL.ENA.
- Step 3. Clear state register: write 0 to register CIMSTATE.

12 AC97/I2S/SPDIF Controller

12.1 Overview

This chapter describes the AIC (AC'97 and I²S Controller) included in this processor.

The AIC supports the Audio Codec '97 Component Specification 2.3 for AC-link format and I2S or IIS (for inter-IC sound), a protocol defined by Philips Semiconductor. Both normal I2S and the MSB-justified I2S formats are supported by AIC.

AIC consists of buffers, status registers, control registers, serializers, and counters for transferring digitized audio between the processor's system memory and an internal I2S CODEC, an external AC97 or I2S CODEC. AIC can record digitized audio by storing the samples in system memory. For playback of digitized audio or production of synthesized audio, the AIC retrieves digitized audio samples from system memory and sends them to a CODEC through the serial connection with AC-link or I2S formats. The internal or external digital-to-analog converter in the CODEC then converts the audio samples into an analog audio waveform. The audio sample data can be stored to and retrieved from system memory either by the DMA controller or by programmed I/O.

The AC-link is a synchronous, fixed-rate serial bus interface for transferring CODEC register control and status information in addition to digital audio. Where both normal I2S and MSB-justified-I2S work with a variety of clock rates, which can be obtained either by dividing the PLL clock by two programmable dividers or from an external clock source.

For I2S systems that support the L3 control bus protocol, additional pins are required to control the external CODEC. CODECs that use an L3 control bus require 3 signals: L3_CLK, L3_DATA, and L3_MODE for writing bytes into the L3 bus register. The AIC supports the L3 bus protocol via software control of the general-purpose I/O (GPIO) pins. The AIC does not provide hardware control for the L3 bus protocol.

To control the internal CODEC, [internal CODEC Spec](#) can be referenced.

SPDIF (Sony/Philips Digital Interconnect Format) is a digital audio interface. The transmission medium can be either electrical or optical. Supports IEC60958 two-channel PCM audio and IEC61937 multi-channel compressed audio (Dolby Digital, DTS, etc.).

This chapter describes the programming model for the AIC. The information in this chapter requires an understanding of the AC'97 specification, Revision 2.3.

12.1.1 Block Diagram

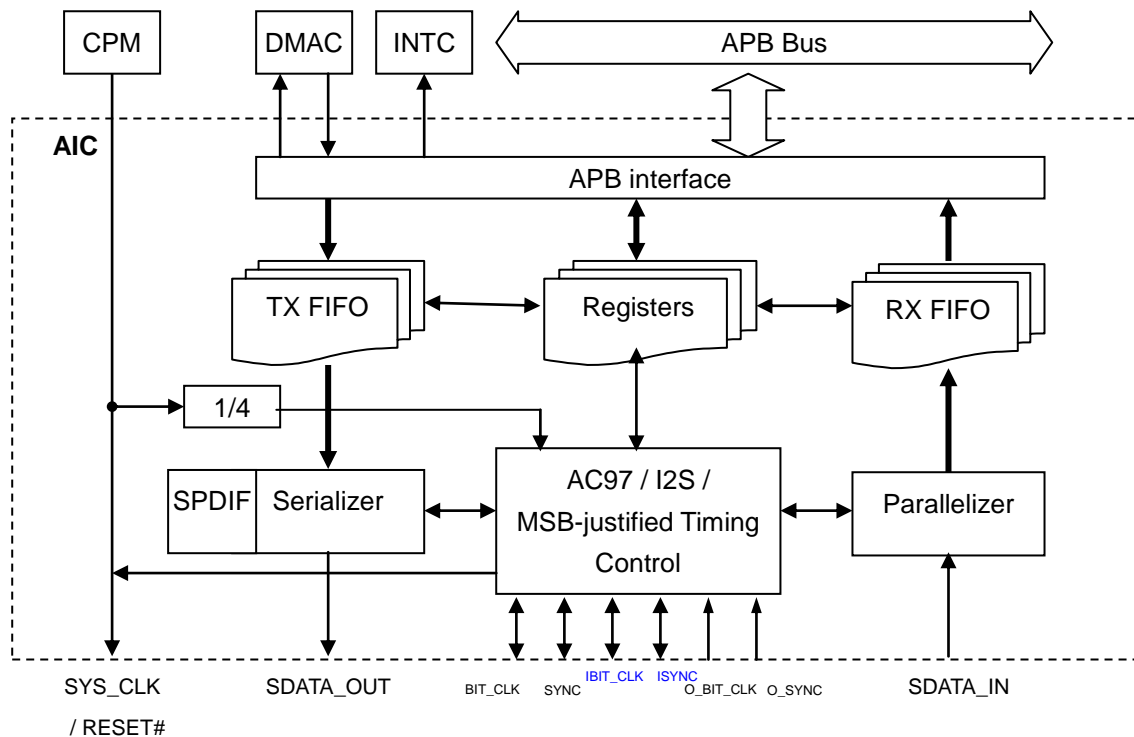


Figure 12-1 AIC Block Diagram

The O_BIT_CLK and O_SYNC ports are only used by inter CODEC.

12.1.2 Features

AIC support following AC97/I2S/SPDIF features:

AC-link (AC97) features

- Up to 20 bit audio sample data sizes supported
- DMA transfer mode supported
- Stop serial clock supported
- Programmable Interrupt function supported
- Support mono PCM data to stereo PCM data expansion on audio play back
- Support endian switch on 16-bits normal audio samples play back
- Support variable sample rate in AC-link format
- Multiple channel output and double rated supported for AC-link format
- Power Down Mode and two Wake-Up modes Supported for AC-link format

I2S features

- 8, 16, 18, 20 and 24 bit audio sample data sizes supported, 16 bits packed sample data is supported
- Up to 8 channels sample data supported

- DMA transfer mode supported
- Stop serial clock supported
- Programmable Interrupt function supported
- Support mono PCM data to stereo PCM data expansion on audio play back
- Support endian switch on 16-bits normal audio samples play back
- Internal programmable or external serial clock and optional system clock supported for I2S or MSB-Justified format
- Internal I2S CODEC supported
- Two FIFOs for transmit and receive respectively
- Support I2S share CLK mode and split CLK mode.

SPDIF features

- 8, 16, 18, 20 and 24 bit audio sample data sizes supported
- DMA transfer mode supported
- Stop serial clock supported
- Programmable Interrupt function supported
- Support IEC60958 two-channel PCM audio
- Support IEC61937 multi-channel compressed audio
- Support consumer mode and only support transmitter mode
- Profession mode is not supported
- The User data bit is '0' as it is not supported in the chip
- Support sampling frequency from 32kHz to 192kHz
- Internal SPDIF HDMI supported

12.1.3 Interface Diagram

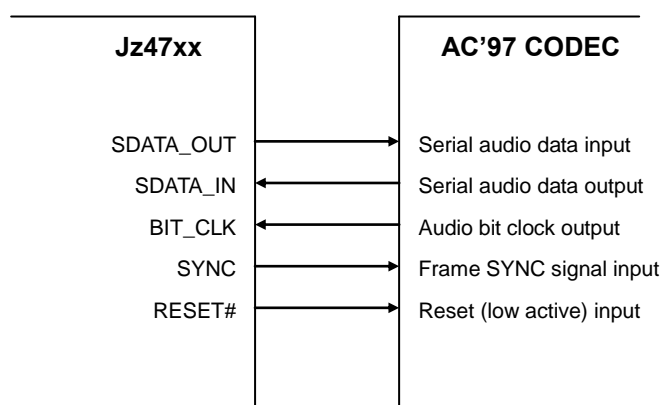
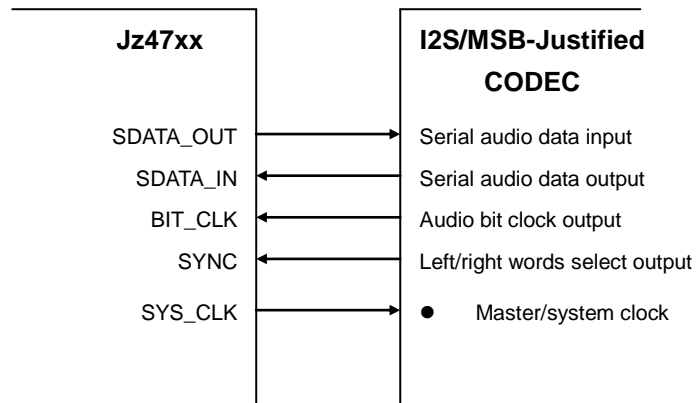
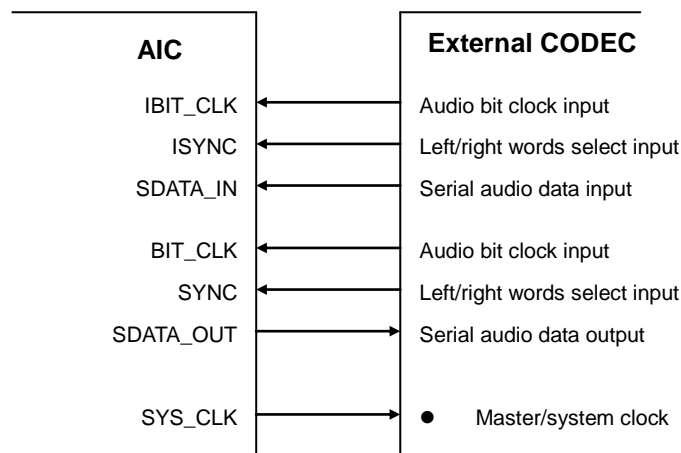


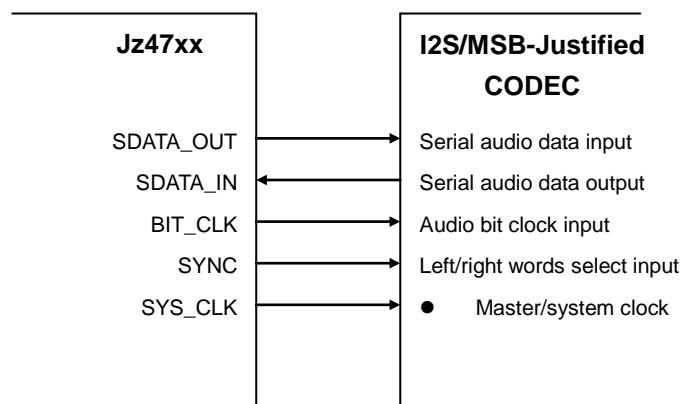
Figure 12-2 Interface to an External AC'97 CODEC Diagram



**Figure 12-3 Interface to an External Master Mode I2S/MSB-Justified CODEC Diagram
(Share Clock Mode)**



**Figure 12-4 Interface to an External Master Mode I2S CODEC Diagram
(Split Clock Mode)**



**Figure 12-5 Interface to an External Slave Mode I2S/MSB-Justified CODEC Diagram
(Share Clock Mode)**

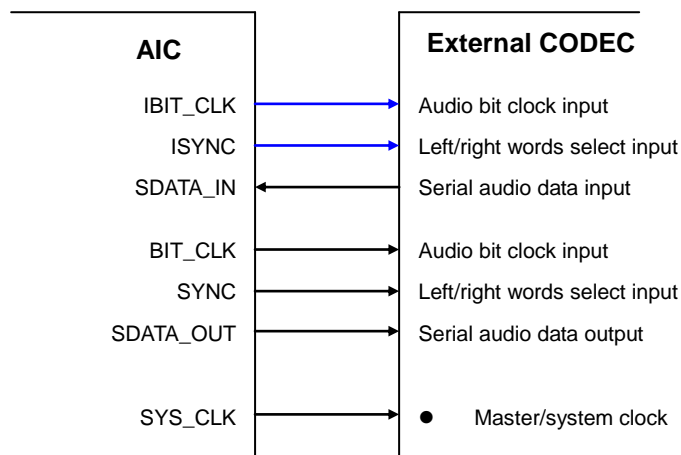


Figure 12-6 Interface to an External Slave Mode I2S CODEC Diagram (Split Clock Mode)

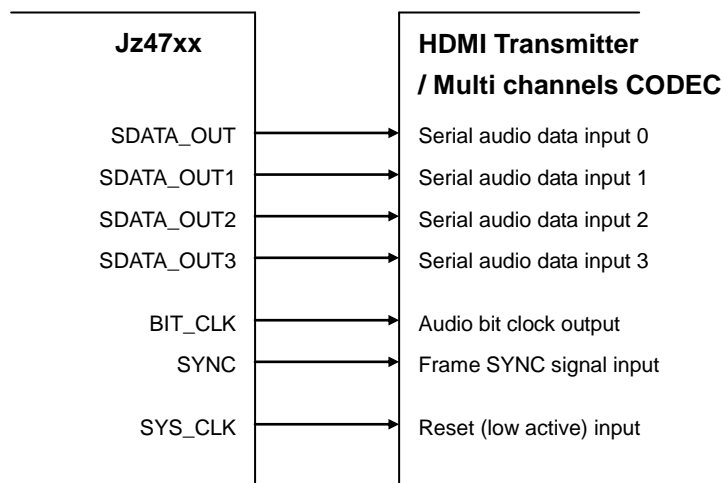


Figure 12-7 Interface to a HDMI Transmitter via I2S Diagram

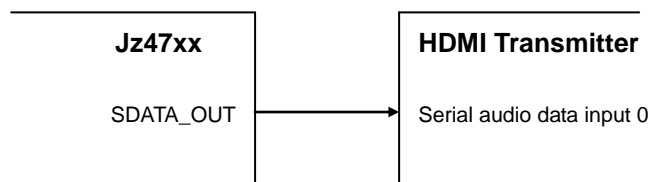


Figure 12-8 Interface to a HDMI Transmitter via SPDIF Diagram

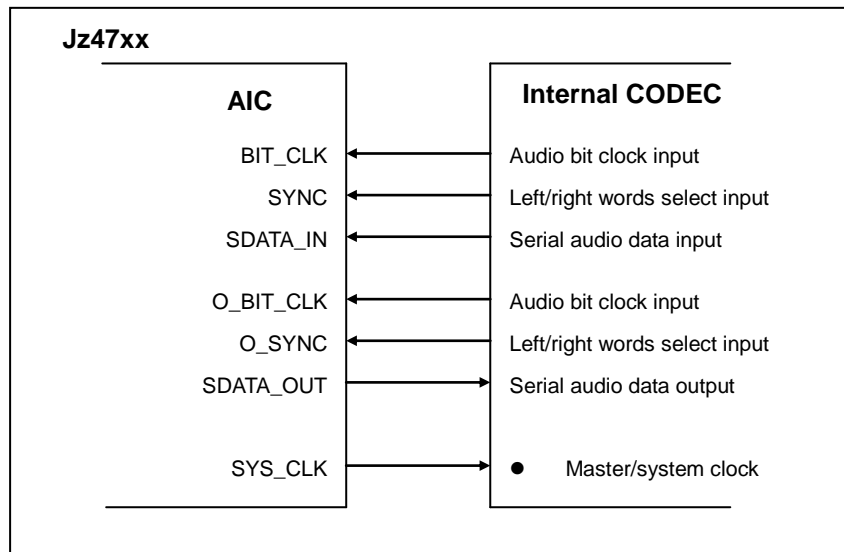


Figure 12-9 Interface to an internal Master Mode I2S CODEC Diagram

Please refer to the related CODEC specification for the details.

12.1.4 Signal Descriptions

There are all 5 pins used to connect between AIC and an external audio CODEC device. If an internal CODEC is used, these pins are not needed. Please refer to [Chip Spec](#). They are listed and described in following table.

Table 12-1 AIC Pins Description

Function Name	PIN Name	I/O	Description
RESET# SYS_CLK	SCLK_R STN	O	RESET#: AC-link format, active-low CODEC reset. SYS_CLK: I2S/MSB-Justified formats, supply system clock to CODEC.
IBIT_CLK	BCLK_A D	I/O	Sample rate dependent bit-rate clock input/output for I2S/MSB-Justified, only use to AD channel.
ISYNC	SYNC_A D	I/O	Indicates the left- or right-channel for I2S/MSB-Justified format, only use to AD channel
SDATA_IN	SDATI	I	Serial audio input data from CODEC.
BIT_CLK	BCLK	I I/O	12.288 MHz bit-rate clock input for AC-link, and sample rate dependent bit-rate clock input/output for I2S/MSB-Justified.
SYNC	SYNC	O	48-kHz frame indicator and synchronizer for AC-link format.
		I/O	Indicates the left- or right-channel for I2S/MSB-Justified format.
SDATA_OUT	SDATO	O	Serial audio output data to CODEC / I2S line 0 / SPDIF output.

SDATA_OUT1	SDATO1	O	Serial audio output data I2S line 1.
SDATA_OUT2	SDATO2	O	Serial audio output data I2S line 2.
SDATA_OUT3	SDATO3	O	Serial audio output data I2S line 3.

The O_BIT_CLK and O_SYNC signals are not connected to any pin for only using by internal CODEC.

12.1.4.1 RESET# / SYS_CLK Pin

RESET# is AC97 active-low CODEC reset, which outputs to CODEC. The CODEC's registers are reset when this RESET# is asserted. This pin is useful only in AC-link format. If AIC is disabled, it retains the high.

SYS_CLK outputs the system clock to CODEC. This pin is useful only in I2S/MSB-justified format. It generates a frequency between approximately 2.048 MHz and 24.576 MHz by dividing down the PLL clock with a programmable divisor. This frequency can be 256, 384, 512 and etc. times of the audio sampling frequency. Or it can be set to a wanted frequency. If AIC is disabled, it retains the high.

12.1.4.2 BIT_CLK Pin

BIT_CLK is the serial data bit rate clock, at which AC97/I2S data moves between the CODEC and the processor. One bit of the serial data is transmitted or received each BIT_CLK period. It is fixed to 12.288 MHz in AC-link format, which inputs from the CODEC. In I2S and MSB-justified format it inputs from the CODEC in slave mode and outputs to CODEC in master mode. In the master mode, the clock is generated internally that is 64 times the sampling frequency. Table 12-8 lists the available sampling frequencies based on an internal clock source. If AIC is disabled, AICFR.AUSEL and AICFR.BCKD determine the direction. And it retains the low if it is output and the state is undefined if it is input.

The IBIT_CLK is for the SDATA_IN signal on division CLOCK function.

12.1.4.3 SYNC Pin

In AC-link format, SYNC provides frame synchronization, fixed to 48kHz, by specifying beginning of an audio sample frame and outputs to CODEC. In I2S/MSB-Justified formats, SYNC is used to indicate left- or right-channel sample data and toggled in sample rate frequency. It outputs to CODEC in master mode and inputs from CODEC in slave mode. If AIC is disabled, AICFR.AUSEL and AICFR.BCKD determine the direction. And it retains the low if it is output and the state is undefined if it is input.

The ISYNC is for the SDATA_IN signal on division CLOCK function.

12.1.4.4 SDATA_OUT Pin

SDATA_OUT is AIC output data pin, which outputs AC97/I2S serial audio data, SPDIF serial data or data of AC97 CODEC register control to an external audio CODEC device.

If in multi channels mode, it outputs the first two channels serial data.
If AIC is disabled, it retains the low.

SDATA_OUTn (n = 1,2,3) is AIC output data pin, which outputs multi channels serial audio data.

12.1.4.5 SDATA_IN Pin

SDATA_IN is AIC inputs data pin, which inputs serial audio data or data of AC97 CODEC register status from an external audio CODEC device. If AIC is disabled, its state is undefined.

12.1.4.6 IBIT_CLK Pin

The IBIT_CLK is for the SDATA_IN signal on division CLOCK function. When in Split Clock mode(AICFR.DMODE = 1), this pin is effective.

12.1.4.7 ISYNC Pin

The ISYNC is for the SDATA_IN signal on division CLOCK function. When in Split Clock mode(AICFR.DMODE = 1), this pin is effective.

12.2 Register Descriptions

AIC software interface includes 13 registers and 1 FIFO data port. They are mapped in IO memory address space so that program can access them to control the operation of AIC and the outside CODEC.

Table 12-2 AIC Registers Description

Name	Description	RW	Reset value	Address	Size
AICFR	AIC Configuration Register	RW	0x07100000	0x10020000	32
AICCR	AIC Common Control Register	RW	0x01240000	0x10020004	32
ACCR1	AIC AC-link Control Register 1	RW	0x00000000	0x10020008	32
ACCR2	AIC AC-link Control Register 2	RW	0x00000000	0x1002000C	32
I2SCR	AIC I2S/MSB-justified Control Register	RW	0x00000000	0x10020010	32
AICSR	AIC FIFO Status Register	RW	0x00000008	0x10020014	32
ACSR	AIC AC-link Status Register	RW	0x00000000	0x10020018	32
I2SSR	AIC I2S/MSB-justified Status Register	RW	0x00000000	0x1002001C	32
ACCAR	AIC AC97 CODEC Command Address Register	RW	0x00000000	0x10020020	32
ACCDR	AIC AC97 CODEC Command Data	RW	0x00000000	0x10020024	32

	Register				
ACSAR	AIC AC97 CODEC Status Address Register	R	0x00000000	0x10020028	32
ACSDR	AIC AC97 CODEC Status Data Register	R	0x00000000	0x1002002C	32
I2SDIV	AIC I2S/MSB-justified Clock Divider Register	RW	0x00000003	0x10020030	32
AICDR	AIC FIFO Data Port Register	RW	0x????????	0x10020034	32
SPENA	SPDIF Enable Register	RW	0x00	0x10020080	8
SPCTRL	SPDIF Control Register	RW	0x0003	0x10020084	16
SPSTATE	SPDIF Status Register	RW	0x0000	0x10020088	16
SPCFG1	SPDIF Configure 1 Register	RW	0x00000000	0x1002008C	32
SPCFG2	SPDIF Configure 2 Register	RW	0x00000000	0x10020090	32
SPFIFO	SPDIF FIFO Register	W	0x????????	0x10020094	32
CKCFG	Clock Configure for the embedded CODEC to AIC	RW	0x00000000 0x00000002	0x100200A0	32
RGADW	Address, data in and write command for accessing to internal registers of embedded CODEC	RW	0x00000000	0x100200A4	32
RGDATA	The read out data and interrupt request status of Internal registers data in the embedded CODEC	R	0x00000000	0x100200A8	32

- 1 AICFR is used to control FIFO threshold, AC-link or I2S/MSB-justified selection, AIC reset, master/slave selection, and AIC enable.
- 2 AICCR is used to control DMA mode, FIFO flush, interrupt enable, internal loop-back, play back and recording enable. It also controls sample size and signed/unsigned data transfer.
- 3 ACCR1 is used to reflect/control valid incoming/outgoing slots of AC97.
- 4 ACCR2 is used to control interrupt enable, output/input sample size, and alternative control of RESET#, SYNC and SDATA_OUT pins in AC-link.
- 5 I2SCR is used to control BIT_CLK stop, audio sample size, I2S or MSB-justified selection in I2S/MSB-justified.
- 6 AICSR is used to reflect FIFOs status.
- 7 ACSR is used to reflect the status of the connected external CODEC in AC-link.
- 8 I2SSR is used to reflect AIC status in I2S/MSB-justified.
- 9 ACCAR and ACCDR are used to hold address and data for AC-link CODEC register read/write.
- 10 ACSAR and ACSDR are used to receive AC-link CODEC registers address and data.
- 11 I2SDIV is used to set clock divider for BIT_CLK generating in I2S/MSB-justified format.
- 12 AICDR is act as data input/output port to/from transmit/receive FIFO when write/read.
- 13 CKCFG, RGADW and RGDATA are used to access internal CODEC, please refer to [CODEC Spec.](#)

12.2.1 AIC Configuration Register (AICFR)

AICFR contains bits to control FIFO threshold, AC-link or I2S/MSB-justified selection, AIC reset, master/slave selection, and AIC enable.

AICFR																0x10020000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				RFTH				Reserved				TFTH				Reserved				IBCKD	ISYNCD	DMODE	Reserved	LSMP	ICDC	AUSEL	RST	BCKD	SYNCD	ENB	
RST	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW						
31:28	Reserved	Writing has no effect, read as zero.	R						
27:24	RFTH	Receive FIFO threshold for interrupt or DMA request. The RFTH valid value is 0 ~ 15. This value represents a threshold value of $(RFTH + 1) * 2$. When the sample number in receive FIFO, indicated by AICSR.RFL, is great than or equal to the threshold value, AICSR.RFS is set. Larger RFTH value provides lower DMA/interrupt request frequency but have more risk to involve receive FIFO overflow. The optimum value is system dependent.	RW						
23:21	Reserved	Writing has no effect, read as zero.	R						
20:16	TFTH	Transmit FIFO threshold for interrupt or DMA request. The TFTH valid value 0 ~ 31. This value represents a threshold value of $TFTH * 2$. When the sample number in transmit FIFO, indicated by AICSR.TFL, is less than or equal to the threshold value, AICSR.TFS is set. Smaller TFTH value provides lower DMA/interrupt request frequency but have more risk to involve transmit FIFO underflow. The optimum value is system dependent.	RW						
15:11	Reserved	Writing has no effect, read as zero.	R						
10	IBCKD	The IBIT_CLK Direction. This bit specifies input/output direction of IBIT_CLK. It is only valid in I2S/MSB-justified format. When AC-link format is selected, IBIT_CLK is always input and this bit is ignored. Change this bit in case of IBIT_CLK is stopped (I2SCR.ISTPBK = 1). This is only available when DMODE = 1. <table border="1" data-bbox="454 1686 1302 1859"> <thead> <tr> <th>BCKD</th> <th>BIT_CLK Direction</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>IBIT_CLK is input from an external source.</td> </tr> <tr> <td>1</td> <td>IBIT_CLK is generated internally and driven out to the CODEC.</td> </tr> </tbody> </table>	BCKD	BIT_CLK Direction	0	IBIT_CLK is input from an external source.	1	IBIT_CLK is generated internally and driven out to the CODEC.	RW
BCKD	BIT_CLK Direction								
0	IBIT_CLK is input from an external source.								
1	IBIT_CLK is generated internally and driven out to the CODEC.								
9	ISYNCD	ISYNC Direction. This bit specifies input/output direction of ISYNC in I2S/MSB-justified format. When AC-link format is selected, ISYNC is always output and this bit is ignored. Change this bit in case of IBIT_CLK is stopped (I2SCR.ISTPBK = 1).	RW						

		<p>This is only available when DMODE = 1.</p> <table border="1"> <thead> <tr> <th>SYNCD</th> <th>SYNC Direction</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>ISYNC is input from an external source.</td> </tr> <tr> <td>1</td> <td>ISYNC is generated internally and driven out to the CODEC.</td> </tr> </tbody> </table>	SYNCD	SYNC Direction	0	ISYNC is input from an external source.	1	ISYNC is generated internally and driven out to the CODEC.	
SYNCD	SYNC Direction								
0	ISYNC is input from an external source.								
1	ISYNC is generated internally and driven out to the CODEC.								
8	DMODE	<p>The Division Clock Mode control.</p> <table border="1"> <thead> <tr> <th>BCKD</th> <th>BIT_CLK Direction</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable. Shared clock mode. The BIT_CLK, SYNC, SDATA_IN and SDATA_OUT are configured to one shared clock I2S channel.</td> </tr> <tr> <td>1</td> <td>Enable. Spilt clock mode. The BIT_CLK, SYNC and SDATA_OUT are configured to one output I2S channel. The IBIT_CLK, ISYNC and SDATA_IN are configured to one input I2S channel.</td> </tr> </tbody> </table>	BCKD	BIT_CLK Direction	0	Disable. Shared clock mode. The BIT_CLK, SYNC, SDATA_IN and SDATA_OUT are configured to one shared clock I2S channel.	1	Enable. Spilt clock mode. The BIT_CLK, SYNC and SDATA_OUT are configured to one output I2S channel. The IBIT_CLK, ISYNC and SDATA_IN are configured to one input I2S channel.	RW
BCKD	BIT_CLK Direction								
0	Disable. Shared clock mode. The BIT_CLK, SYNC, SDATA_IN and SDATA_OUT are configured to one shared clock I2S channel.								
1	Enable. Spilt clock mode. The BIT_CLK, SYNC and SDATA_OUT are configured to one output I2S channel. The IBIT_CLK, ISYNC and SDATA_IN are configured to one input I2S channel.								
7	Reserved	Writing has no effect, read as zero.	R						
6	LSMP	<p>Select between play last sample or play ZERO sample in TX FIFO underflow. ZERO sample means sample value is zero. This bit is better be changed while audio replay is stopped.</p> <table border="1"> <thead> <tr> <th>LSMP</th> <th>CODEC used</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Play ZERO sample when TX FIFO underflow.</td> </tr> <tr> <td>1</td> <td>Play last sample when TX FIFO underflow.</td> </tr> </tbody> </table>	LSMP	CODEC used	0	Play ZERO sample when TX FIFO underflow.	1	Play last sample when TX FIFO underflow.	RW
LSMP	CODEC used								
0	Play ZERO sample when TX FIFO underflow.								
1	Play last sample when TX FIFO underflow.								
5	ICDC	<p>Internal CODEC used. Select between internal or external CODEC.</p> <table border="1"> <thead> <tr> <th>ICDC</th> <th>CODEC used</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>External CODEC.</td> </tr> <tr> <td>1</td> <td>Internal CODEC.</td> </tr> </tbody> </table>	ICDC	CODEC used	0	External CODEC.	1	Internal CODEC.	RW
ICDC	CODEC used								
0	External CODEC.								
1	Internal CODEC.								
4	AUSEL	<p>Audio Unit Select. Select between AC-link and I2S/MSB-justified. Change this bit in case of BIT_CLK is stopped (I2SCR.STPBK = 1).</p> <table border="1"> <thead> <tr> <th>AUSEL</th> <th>Selected</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Select AC-link format.</td> </tr> <tr> <td>1</td> <td>Select I2S/MSB-justified format.</td> </tr> </tbody> </table>	AUSEL	Selected	0	Select AC-link format.	1	Select I2S/MSB-justified format.	RW
AUSEL	Selected								
0	Select AC-link format.								
1	Select I2S/MSB-justified format.								
3	RST	Reset AIC. Write 1 to this bit reset AIC registers and FIFO except AICFR and I2SDIV register. Writing 0 to this bit has no effect and this bit is always reading 0.	W						
2	BCKD	<p>BIT_CLK Direction. This bit specifies input/output direction of BIT_CLK. It is only valid in I2S/MSB-justified format. When AC-link format is selected, BIT_CLK is always input and this bit is ignored. Change this bit in case of BIT_CLK is stopped (I2SCR.STPBK = 1).</p> <table border="1"> <thead> <tr> <th>BCKD</th> <th>BIT_CLK Direction</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>BIT_CLK is input from an external source.</td> </tr> <tr> <td>1</td> <td>BIT_CLK is generated internally and driven out to</td> </tr> </tbody> </table>	BCKD	BIT_CLK Direction	0	BIT_CLK is input from an external source.	1	BIT_CLK is generated internally and driven out to	RW
BCKD	BIT_CLK Direction								
0	BIT_CLK is input from an external source.								
1	BIT_CLK is generated internally and driven out to								

			the CODEC.							
1	SYNCD	SYNC Direction. This bit specifies input/output direction of SYNC in I2S/MSB-justified format. When AC-link format is selected, SYNC is always output and this bit is ignored. Change this bit in case of BIT_CLK is stopped (I2SCR.STPBK = 1).		RW						
		<table border="1"> <thead> <tr> <th>SYNCD</th> <th>SYNC Direction</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>SYNC is input from an external source.</td> </tr> <tr> <td>1</td> <td>SYNC is generated internally and driven out to the CODEC.</td> </tr> </tbody> </table>	SYNCD	SYNC Direction	0	SYNC is input from an external source.	1	SYNC is generated internally and driven out to the CODEC.		
SYNCD	SYNC Direction									
0	SYNC is input from an external source.									
1	SYNC is generated internally and driven out to the CODEC.									
0	ENB	Enable AIC function. This bit is used to enable or disable the AIC function.		RW						
		<table border="1"> <thead> <tr> <th>ENB</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable AIC Controller.</td> </tr> <tr> <td>1</td> <td>Enable AIC Controller.</td> </tr> </tbody> </table>	ENB	Description	0	Disable AIC Controller.	1	Enable AIC Controller.		
ENB	Description									
0	Disable AIC Controller.									
1	Enable AIC Controller.									

The different I2S mode configurations are listed and described in following table

Table 12-3 AIC I2S mode configurations

ICDC	BCKD	SYNC D	IBCK D	ISYNC D	DMOD E	Description
1	0	0	0	0	X	Internal CODEC I2S mode.
0	0	0	0	0	1	External CODEC Spilt clock I2S/MSB-justified master mode.
0	1	1	1	1	1	External CODEC Spilt clock I2S/MSB-justified slave mode.
0	0	0	1	1	1	External CODEC Spilt clock I2S/MSB-justified DAC master mode and I2S/MSB-justified ADC slave mode.
0	1	1	0	0	1	External CODEC Spilt clock I2S/MSB-justified ADC master mode and I2S/MSB-justified DAC slave mode.
0	1	1	X	X	0	External CODEC share clock I2S/MSB-justified slave mode.
0	0	0	X	X	0	External CODEC share clock I2S/MSB-justified master mode.
0	0	1	X	X	0	AIC roles the master with external serial clock source of I2S/MSB-justified interface.

12.2.2 AIC Common Control Register (AICCR)

AICCR contains bits to control DMA mode, FIFO flush, interrupt enable, internal loop-back, play back and recording enable. It also controls sample size and signed/unsigned data transfer.

AICCR																0x10020004																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved	Reserved	PACK16	Reserved	CHANNEL	Reserved	OSS	ISS	RDMS	TDMS	Reserved	M2S	ENDSW	ASVTSU	TFLUSH	RFLUSH	EROR	ETUR	ERFS	ETFS	ENLBF	ERPL	EREC										
RST	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW																		
31:30	Reserved	Writing has no effect, read as zero.	R																		
29	Reserved	Keep this value to 0 in normal use.	R																		
28	PACK16	Output Sample data 16bit packed mode select. This bit reflects that one word contains two sample data or only one sample data with LSB align. The packed mode is only support 16bit sample size. <table border="1" data-bbox="550 981 1249 1234"> <thead> <tr> <th>PACK16</th> <th>Sample Size</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Unpacked data mode. One word only contains one 16bit sample data aligned LSB.</td> </tr> <tr> <td>1</td> <td>Packed data mode. One word contains two 16 bit sample data.</td> </tr> </tbody> </table>	PACK16	Sample Size	0	Unpacked data mode. One word only contains one 16bit sample data aligned LSB.	1	Packed data mode. One word contains two 16 bit sample data.	RW												
PACK16	Sample Size																				
0	Unpacked data mode. One word only contains one 16bit sample data aligned LSB.																				
1	Packed data mode. One word contains two 16 bit sample data.																				
27	Reserved	Writing has no effect, read as zero.	R																		
26:24	CHANNEL	Output Channel Number Select. These bits reflect output data channels from AIC to device. The data supported are: 1(mono), 2(stereo), 4, 6 and 8 channels. The sample data is LSB-justified in memory/register. <table border="1" data-bbox="550 1402 1249 1789"> <thead> <tr> <th>CHANNEL</th> <th>Sample Size</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>1 channel, mono, Only SDATA0 used.</td> </tr> <tr> <td>0x1</td> <td>2 channels, stereo, Only SDATA0 used.</td> </tr> <tr> <td>0x2</td> <td>Reserved.</td> </tr> <tr> <td>0x3</td> <td>4 channels, SDATA0 and SDATA1 used.</td> </tr> <tr> <td>0x4</td> <td>Reserved.</td> </tr> <tr> <td>0x5</td> <td>6 channels, SDATA0 to SDATA2 used.</td> </tr> <tr> <td>0x6</td> <td>Reserved.</td> </tr> <tr> <td>0x7</td> <td>8 channels, , SDATA0 to SDATA3 used.</td> </tr> </tbody> </table>	CHANNEL	Sample Size	0x0	1 channel, mono, Only SDATA0 used.	0x1	2 channels, stereo, Only SDATA0 used.	0x2	Reserved.	0x3	4 channels, SDATA0 and SDATA1 used.	0x4	Reserved.	0x5	6 channels, SDATA0 to SDATA2 used.	0x6	Reserved.	0x7	8 channels, , SDATA0 to SDATA3 used.	RW
CHANNEL	Sample Size																				
0x0	1 channel, mono, Only SDATA0 used.																				
0x1	2 channels, stereo, Only SDATA0 used.																				
0x2	Reserved.																				
0x3	4 channels, SDATA0 and SDATA1 used.																				
0x4	Reserved.																				
0x5	6 channels, SDATA0 to SDATA2 used.																				
0x6	Reserved.																				
0x7	8 channels, , SDATA0 to SDATA3 used.																				
23:22	Reserved	Writing has no effect, read as zero.	R																		
21:19	OSS	Output Sample Size. These bits reflect output sample data size from memory or register. The data sizes supported are: 8, 16, 18, 20 and 24 bits. The sample data is LSB-justified in memory/register.	RW																		

			<table border="1"> <thead> <tr> <th>OSS</th> <th>Sample Size</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>8 bit.</td> </tr> <tr> <td>0x1</td> <td>16 bit.</td> </tr> <tr> <td>0x2</td> <td>18 bit.</td> </tr> <tr> <td>0x3</td> <td>20 bit.</td> </tr> <tr> <td>0x4</td> <td>24 bit.</td> </tr> <tr> <td>0x5~0x7</td> <td>Reserved.</td> </tr> </tbody> </table>	OSS	Sample Size	0x0	8 bit.	0x1	16 bit.	0x2	18 bit.	0x3	20 bit.	0x4	24 bit.	0x5~0x7	Reserved.	
OSS	Sample Size																	
0x0	8 bit.																	
0x1	16 bit.																	
0x2	18 bit.																	
0x3	20 bit.																	
0x4	24 bit.																	
0x5~0x7	Reserved.																	
18:16	ISS	<p>Input Sample Size. These bits reflect input sample data size to memory or register. The data sizes supported are: 8, 16, 18, 20 and 24 bits. The sample data is LSB-justified in memory/register.</p> <table border="1"> <thead> <tr> <th>ISS</th> <th>Sample Size</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>8 bit.</td> </tr> <tr> <td>0x1</td> <td>16 bit.</td> </tr> <tr> <td>0x2</td> <td>18 bit.</td> </tr> <tr> <td>0x3</td> <td>20 bit.</td> </tr> <tr> <td>0x4</td> <td>24 bit.</td> </tr> <tr> <td>0x5~0x7</td> <td>Reserved.</td> </tr> </tbody> </table>	ISS	Sample Size	0x0	8 bit.	0x1	16 bit.	0x2	18 bit.	0x3	20 bit.	0x4	24 bit.	0x5~0x7	Reserved.	RW	
ISS	Sample Size																	
0x0	8 bit.																	
0x1	16 bit.																	
0x2	18 bit.																	
0x3	20 bit.																	
0x4	24 bit.																	
0x5~0x7	Reserved.																	
15	RDMS	<p>Receive DMA enable. This bit is used to enable or disable the DMA during receiving audio data.</p> <table border="1"> <thead> <tr> <th>RDMS</th> <th>Receive DMA</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </tbody> </table>	RDMS	Receive DMA	0	Disabled.	1	Enabled.	RW									
RDMS	Receive DMA																	
0	Disabled.																	
1	Enabled.																	
14	TDMS	<p>Transmit DMA enable. This bit is used to enable or disable the DMA during transmit audio data.</p> <table border="1"> <thead> <tr> <th>TDMS</th> <th>Transmit DMA</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </tbody> </table>	TDMS	Transmit DMA	0	Disabled.	1	Enabled.	RW									
TDMS	Transmit DMA																	
0	Disabled.																	
1	Enabled.																	
13:12	Reserved	Writing has no effect, read as zero.	R															
11	M2S	<p>Mono To Stereo. This bit control whether to do mono to stereo sample expansion in play back. When this bit is set, every outgoing sample data in the steam plays in both left and right channels. This bit should only be set in 2 channels configuration. It takes effective immediately when the bit is changed. Change this before replay started.</p> <table border="1"> <thead> <tr> <th>M2S</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No mono to stereo expansion.</td> </tr> <tr> <td>1</td> <td>Do mono to stereo expansion.</td> </tr> </tbody> </table>	M2S	Description	0	No mono to stereo expansion.	1	Do mono to stereo expansion.	RW									
M2S	Description																	
0	No mono to stereo expansion.																	
1	Do mono to stereo expansion.																	
10	ENDSW	<p>Endian Switch. This bit control endian change on outgoing 16-bits size audio sample by swapping high and low bytes in the sample data.</p> <table border="1"> <thead> <tr> <th>ENDSW</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No change on outgoing sample</td> </tr> </tbody> </table>	ENDSW	Description	0	No change on outgoing sample	RW											
ENDSW	Description																	
0	No change on outgoing sample																	

				data.							
			1	Swap high and low byte for outgoing 16-bits size sample data.							
9	ASVTSU	<p>Audio Sample Value Transfer between Signed and Unsigned data format. This bit is used to control the signed \leftrightarrow unsigned data transfer. If it is 1, the incoming and outgoing audio sample data will be transferred by toggle its most significant bit.</p> <table border="1"> <thead> <tr> <th>ASVTSU</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No audio sample value signed \leftrightarrow unsigned transfer.</td> </tr> <tr> <td>1</td> <td>Do audio sample value signed \leftrightarrow unsigned transfer.</td> </tr> </tbody> </table>			ASVTSU	Description	0	No audio sample value signed \leftrightarrow unsigned transfer.	1	Do audio sample value signed \leftrightarrow unsigned transfer.	RW
ASVTSU	Description										
0	No audio sample value signed \leftrightarrow unsigned transfer.										
1	Do audio sample value signed \leftrightarrow unsigned transfer.										
8	TFLUSH	Transmit FIFO Flush. Write 1 to this bit flush transmit FIFOs to empty. Writing 0 to this bit has no effect and this bit is always reading 0.			W						
7	RFLUSH	Receive FIFO Flush. Write 1 to this bit flush receive FIFOs to empty. Writing 0 to this bit has no effect and this bit is always reading 0.			W						
6	EROR	<p>Enable ROR Interrupt. This bit is used to control the ROR interrupt enable or disable.</p> <table border="1"> <thead> <tr> <th>EROR</th> <th>ROR Interrupt</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </tbody> </table>			EROR	ROR Interrupt	0	Disabled.	1	Enabled.	RW
EROR	ROR Interrupt										
0	Disabled.										
1	Enabled.										
5	ETUR	<p>Enable TUR Interrupt. This bit is used to control the TUR interrupt enable or disable.</p> <table border="1"> <thead> <tr> <th>ETUR</th> <th>TUR Interrupt</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </tbody> </table>			ETUR	TUR Interrupt	0	Disabled.	1	Enabled.	RW
ETUR	TUR Interrupt										
0	Disabled.										
1	Enabled.										
4	ERFS	<p>Enable RFS Interrupt. This bit is used to control the RFS interrupt enable or disable.</p> <table border="1"> <thead> <tr> <th>ERFS</th> <th>RFS Interrupt</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </tbody> </table>			ERFS	RFS Interrupt	0	Disabled.	1	Enabled.	RW
ERFS	RFS Interrupt										
0	Disabled.										
1	Enabled.										
3	ETFS	<p>Enable TFS Interrupt. This bit is used to control the TFS interrupt enable or disable.</p> <table border="1"> <thead> <tr> <th>ETFS</th> <th>TFS Interrupt</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </tbody> </table>			ETFS	TFS Interrupt	0	Disabled.	1	Enabled.	RW
ETFS	TFS Interrupt										
0	Disabled.										
1	Enabled.										
2	ENLBF	<p>Enable AIC Loop Back Function. This bit is used to enable or disable the internal loop back function of AIC, which is used for test only. When the AIC loop back function is enabled, normal audio replay/record functions are disabled.</p> <table border="1"> <thead> <tr> <th>ENLBF</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>AIC Loop Back Function is Disabled.</td> </tr> </tbody> </table>			ENLBF	Description	0	AIC Loop Back Function is Disabled.	RW		
ENLBF	Description										
0	AIC Loop Back Function is Disabled.										

		1	AIC Loop Back Function is Enabled.	
1	ERPL	Enable Playing Back function. This bit is used to disable or enable the audio sample data transmitting.		RW
		ERPL	Description	
		0	AIC Playing Back Function is Disabled.	
		1	AIC Playing Back Function is Enabled.	
0	EREC	Enable Recording Function. This bit is used to disable or enable the audio sample data receiving.		RW
		EREC	Description	
		0	AIC Recording Function is Disabled.	
		1	AIC Recording Function is Enabled.	

12.2.3 AIC AC-link Control Register 1 (ACCR1)

ACCR1 contains bits to reflect/control valid incoming/outgoing slots of AC97. It is used only in AC-link format.

ACCR1																0x10020008																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved						RS						Reserved						XS													
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW						
31:26	Reserved	Writing has no effect, read as zero.	R						
25:16	RS	Receive Valid Slots. These bits are used to indicate which incoming slots are valid. Slot 3 is mapped to bit 16 or RS[0], slot 4 to bit 17 or RS[1] and so on. When write to this field, a bit 1 means we expect a PCM data in the corresponding slot, a bit 0 means the corresponding slot PCM data will be discarded. When read from this field, a bit 1 means we receive an expected valid PCM data in the corresponding slot. This field should be written before record started. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>RS[n] Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Slot n+3 is invalid.</td> </tr> <tr> <td>1</td> <td>Slot n+3 has valid PCM data.</td> </tr> </tbody> </table>	RS[n] Value	Description	0	Slot n+3 is invalid.	1	Slot n+3 has valid PCM data.	RW
RS[n] Value	Description								
0	Slot n+3 is invalid.								
1	Slot n+3 has valid PCM data.								
15:10	Reserved	Writing has no effect, read as zero.	R						
9:0	XS	Transmit Valid Slots. These bits making up slots map to the valid bits in the AC'97 tag (slot 0 on SDATA_OUT) and indicate which outgoing slots have valid PCM data. Bit 0 or XS[0] maps to slot 3, bit 1 or XS[1] to slot 4 and so on. Setting the corresponding bit indicates to AIC to take an audio sample from transmit FIFO to fill the respective slot. And it indicates to	RW						

		the CODEC that valid PCM data will be in the respective slot. The number of valid bits will designate how many words will be pulled out of the FIFO per audio frame. This field should be written before record and replay started.							
		<table border="1"> <thead> <tr> <th>XS[n] Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Slot n+3 is invalid.</td> </tr> <tr> <td>1</td> <td>Slot n+3 has valid PCM data.</td> </tr> </tbody> </table>	XS[n] Value	Description	0	Slot n+3 is invalid.	1	Slot n+3 has valid PCM data.	
XS[n] Value	Description								
0	Slot n+3 is invalid.								
1	Slot n+3 has valid PCM data.								

12.2.4 AIC AC-link Control Register 2 (ACCR2)

ACCR2 contains bits to control interrupt enable, output/input sample size, and alternative control of RESET#, SYNC and SDATA_OUT pins in AC-link. It is valid only in AC-link format.

	ACCR2																								0x1002000C								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved													ERSTO	ESADR	ECADT	Reserved										SO	SR	SS	SA			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW						
31:19	Reserved	Writing has no effect, read as zero.	R						
18	ERSTO	Enable RSTO Interrupt. This bit is used to control the RSTO interrupt enable or disable. <table border="1"> <thead> <tr> <th>ERSTO</th> <th>RSTO Interrupt</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </tbody> </table>	ERSTO	RSTO Interrupt	0	Disabled.	1	Enabled.	RW
ERSTO	RSTO Interrupt								
0	Disabled.								
1	Enabled.								
17	ESADR	Enable SADR Interrupt. This bit is used to control the SADR interrupt enable or disable. <table border="1"> <thead> <tr> <th>ESADR</th> <th>SADR Interrupt</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </tbody> </table>	ESADR	SADR Interrupt	0	Disabled.	1	Enabled.	RW
ESADR	SADR Interrupt								
0	Disabled.								
1	Enabled.								
16	ECADT	Enable CADT Interrupt. This bit is used to control the CADT interrupt enable or disable. <table border="1"> <thead> <tr> <th>ECADT</th> <th>CADT Interrupt</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </tbody> </table>	ECADT	CADT Interrupt	0	Disabled.	1	Enabled.	RW
ECADT	CADT Interrupt								
0	Disabled.								
1	Enabled.								
15:4	Reserved	Writing has no effect, read as zero.	R						
3	SO	SDATA_OUT output value. When SA is 1, this bit controls SDATA_OUT pin voltage level, 0 for low, 1 for high; otherwise, it is ignored.	RW						
2	SR	RESET# pin level. When AC-link is selected, this bit is used to drive the RESET# pin.	RW						

			SR	RESET# Pin Voltage Level	
			0	High.	
			1	Low.	
1	SS	SYNC value. When this bit is read, it returns the actual value of SYNC. When SA is 1, write value controls SYNC pin value. When SA is 0, write to it is ignored.			RW
0	SA	SYNC and SDATA_OUT Alternation. This bit is used to determine the driven signal of SYNC and SDATA_OUT. When SA is 0, SYNC and SDATA_OUT being driven AIC function logic; otherwise, SYNC is controlled by the SS and SDATA_OUT is controlled by the SO. The true table of SYNC is described in following.			RW
			SA	SS	Description
		0	0	When read, indicated SYNC is 0.	
				When write, not effect.	
		1	1	When read, indicated SYNC is 1.	
				When write, not effect.	
		1	0	When read, indicated SYNC is 0.	
				When write, SYNC is driven to 0.	
		1	1	When read, indicated SYNC is 1.	
				When write, SYNC is driven to 1.	

12.2.5 AIC I2S/MSB-justified Control Register (I2SCR)

I2SCR contains bits to control BIT_CLK stop, audio sample size, I2S or MSB-justified selection in I2S/MSB-justified. It is valid only in I2S/MSB-justified format.

I2SCR																	0x10020010															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														RFIRST	SWLH	Reserved	ISTPBK	STPBK	Reserved							ESCLK	Reserved			AMSL	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:18	Reserved	Writing has no effect, read as zero.	R
17	RFIRST	Send R channel first in stereo mode. This bit should only be set in 2 channels configuration. The frame is LR like or RL like. It takes effective immediately when the bit is changed. Change this before replay started.	RW
		RFIRST	Description
		0	Send L channel first. (LR)
		1	Send R channel first. (RL)

16	SWLH	<p>Switch LR channel in 16bit-packed stereo mode.</p> <p>This bit control whether the low address data is L or R. This bit should only be set in 2 channels configuration and 16bit-packed mode. That means it can only valid with packed mode (PACK16 =1) and 2 channels (CHANNEL=0x1).</p> <p>It takes effective immediately when the bit is changed. Change this before replay started.</p> <table border="1"> <thead> <tr> <th>SWLH</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>16 bit LSB and 16bit MSB is not switched.</td> </tr> <tr> <td>1</td> <td>16 bit LSB and 16bit MSB is switched.</td> </tr> </tbody> </table>	SWLH	Description	0	16 bit LSB and 16bit MSB is not switched.	1	16 bit LSB and 16bit MSB is switched.	RW
SWLH	Description								
0	16 bit LSB and 16bit MSB is not switched.								
1	16 bit LSB and 16bit MSB is switched.								
15:12	Reserved	Writing has no effect, read as zero.	R						
13	ISTPBK	<p>Stop IBIT_CLK. It is used to stop the IBIT_CLK in I2S/MSB-justified format. When AC-link is selected, all of its operations are ignored.</p> <table border="1"> <thead> <tr> <th>STPBK</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>IBIT_CLK is not stopped.</td> </tr> <tr> <td>1</td> <td>IBIT_CLK is stopped.</td> </tr> </tbody> </table> <p>Please set this bit to 1 to stop IBIT_CLK when change AICFR.AUSEL and AICFR.IBCKD.</p>	STPBK	Description	0	IBIT_CLK is not stopped.	1	IBIT_CLK is stopped.	RW
STPBK	Description								
0	IBIT_CLK is not stopped.								
1	IBIT_CLK is stopped.								
12	STPBK	<p>Stop BIT_CLK. It is used to stop the BIT_CLK in I2S/MSB-justified format. When AC-link is selected, all of its operations are ignored.</p> <table border="1"> <thead> <tr> <th>STPBK</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>BIT_CLK is not stopped.</td> </tr> <tr> <td>1</td> <td>BIT_CLK is stopped.</td> </tr> </tbody> </table> <p>Please set this bit to 1 to stop BIT_CLK when change AICFR.AUSEL and AICFR.BCKD.</p>	STPBK	Description	0	BIT_CLK is not stopped.	1	BIT_CLK is stopped.	RW
STPBK	Description								
0	BIT_CLK is not stopped.								
1	BIT_CLK is stopped.								
11:5	Reserved	Writing has no effect, read as zero.	R						
4	ESCLK	Enable SYSCLK output. When this bit is 1, the SYSCLK outputs to chip outside is enabled. Else, the clock is disabled.	RW						
3:1	Reserved	Writing has no effect, read as zero.	R						
0	AMSL	<p>Specify Alternate Mode (I2S or MSB-Justified) Operation.</p> <table border="1"> <thead> <tr> <th>AMSL</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Select I2S Operation Mode.</td> </tr> <tr> <td>1</td> <td>Select MSB-Justified Operation Mode.</td> </tr> </tbody> </table>	AMSL	Description	0	Select I2S Operation Mode.	1	Select MSB-Justified Operation Mode.	RW
AMSL	Description								
0	Select I2S Operation Mode.								
1	Select MSB-Justified Operation Mode.								

12.2.6 AIC Controller FIFO Status Register (AICSR)

AICSR contains bits to reflect FIFOs status. Most of the bits are read-only except two, which can be written a 0.

AICSR																0x10020014																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved		RFL				Reserved								TFL				Reserved	ROR	TUR	RFS	TFS	Reserved									
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW						
31:30	Reserved	Writing has no effect, read as zero.	R						
29:24	RFL	Receive FIFO Level. The bits indicate the amount of valid PCM data in Receive FIFO. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>RFL Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x00 ~ 0x20</td> <td>RFL valid PCM data in receive FIFO.</td> </tr> <tr> <td>0x21 ~ 0x3F</td> <td>Reserved.</td> </tr> </tbody> </table>	RFL Value	Description	0x00 ~ 0x20	RFL valid PCM data in receive FIFO.	0x21 ~ 0x3F	Reserved.	R
RFL Value	Description								
0x00 ~ 0x20	RFL valid PCM data in receive FIFO.								
0x21 ~ 0x3F	Reserved.								
23:14	Reserved	Writing has no effect, read as zero.	R						
13:8	TFL	Transmit FIFO Level. The bits indicate the amount of valid PCM data in Transmit FIFO. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>TFL Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x00 ~ 0x3F</td> <td>TFL valid PCM data in transmit FIFO.</td> </tr> </tbody> </table>	TFL Value	Description	0x00 ~ 0x3F	TFL valid PCM data in transmit FIFO.	R		
TFL Value	Description								
0x00 ~ 0x3F	TFL valid PCM data in transmit FIFO.								
7	Reserved	Writing has no effect, read as zero.	R						
6	ROR	Receive FIFO Over Run. This bit indicates that receive FIFO has or has not experienced an overrun. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>ROR</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>When read, indicates over-run has not been found. When write, clear itself.</td> </tr> <tr> <td>1</td> <td>When read, indicates data has even been written to full receive FIFO. When write, not effects.</td> </tr> </tbody> </table>	ROR	Description	0	When read, indicates over-run has not been found. When write, clear itself.	1	When read, indicates data has even been written to full receive FIFO. When write, not effects.	RW
ROR	Description								
0	When read, indicates over-run has not been found. When write, clear itself.								
1	When read, indicates data has even been written to full receive FIFO. When write, not effects.								
5	TUR	Transmit FIFO Under Run. This bit indicates that transmit FIFO has or has not experienced an under-run. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>TUR</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>When read, indicates under-run has not been found. When write, clear itself.</td> </tr> <tr> <td>1</td> <td>When read, indicates data has even been read from empty transmit FIFO. When write, not effects.</td> </tr> </tbody> </table>	TUR	Description	0	When read, indicates under-run has not been found. When write, clear itself.	1	When read, indicates data has even been read from empty transmit FIFO. When write, not effects.	RW
TUR	Description								
0	When read, indicates under-run has not been found. When write, clear itself.								
1	When read, indicates data has even been read from empty transmit FIFO. When write, not effects.								
4	RFS	Receive FIFO Service Request. This bit indicates that receive FIFO level is or not below receive FIFO threshold, which is controlled by AICFR.RFTH. When RFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>RFS</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	RFS	Description			R		
RFS	Description								

			0	Receive FIFO level below RFL threshold.		
			1	Receive FIFO level at or above RFL threshold.		
3	TFS	Transmit FIFO Service Request. This bit indicates that transmit FIFO level is below Transmit FIFO threshold, which is controlled by AICFR.TFTH. When TFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting.				R
			TFS	Description		
			0	Transmit FIFO level exceeds TFL threshold.		
			1	Transmit FIFO level at or below TFL threshold.		
2:0	Reserved	Writing has no effect, read as zero.				R

12.2.7 AIC AC-link Status Register (ACSR)

ACSR contains bits to reflect the status of the connected external CODEC in AC-link format. Bits in this register are read-only in general, except some of them can be written a 0.

	ACSR																0x10020018															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved										SLTERR	CRDY	CLPM	RSTO	SADR	CADT	Reserved															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW						
31:22	Reserved	Writing has no effect, read as zero.	R						
21	SLTERR	Hardware detects a Slot Error. This bit indicates an error in SLOTREQ bits on incoming data from external CODEC is detected. The error can be: (1) find 1 in a SLOTREQ bit, which corresponding to an inactive slot; (2) all active slots should be request in the same time by SLOTREQ, but an exception is found. All errors are accumulated to ACSR.SLTERR by hardware until software clears it. Software writes 0 clear this bit and write 1 has no effect.	RW						
20	CRDY	External CODEC Ready. This bit is derived from the CODEC Ready bit of Slot 0 in SDATA_IN, and it indicates the external AC97 CODEC is ready or not.	R						
		<table border="1"> <thead> <tr> <th>CRDY</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CODEC is not ready.</td> </tr> <tr> <td>1</td> <td>CODEC is ready.</td> </tr> </tbody> </table>	CRDY	Description	0	CODEC is not ready.	1	CODEC is ready.	
CRDY	Description								
0	CODEC is not ready.								
1	CODEC is ready.								
19	CLPM	External CODEC Low Power Mode. This bit indicates the external CODEC is switched to low power mode or BIT_CLK is active from CODEC after wake up.	R						
		<table border="1"> <thead> <tr> <th>CLPM</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	CLPM	Description					
CLPM	Description								

			0	BIT_CLK is active.								
			1	CODEC is switched to low power mode.								
18	RSTO	External CODEC Registers Read Status Time Out. This bit indicates that the read status time out is detected or not. It is set to 1 if the data not return in 4 frames after a CODEC registers read command issued.	<table border="1"> <thead> <tr> <th>RSTO</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>When read, indicates time out has not occurred.</td> </tr> <tr> <td>1</td> <td>When read, indicates read status time out found.</td> </tr> </tbody> </table>		RSTO	Description	0	When read, indicates time out has not occurred.	1	When read, indicates read status time out found.	RW	
RSTO	Description											
0	When read, indicates time out has not occurred.											
1	When read, indicates read status time out found.											
		Write 0 clear this bit and write 1 is ignored. When RSTO is 1, it may trigger an interrupt depends on the interrupt enable setting.										
17	SADR	External CODEC Registers Status Address and Data Received. This bit indicates that address and data of an external AC '97 CODEC register has or has not been received.	<table border="1"> <thead> <tr> <th>SADR</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>When read, indicates no register address/data received.</td> </tr> <tr> <td>1</td> <td>When read, indicates address/data received.</td> </tr> </tbody> </table>		SADR	Description	0	When read, indicates no register address/data received.	1	When read, indicates address/data received.	RW	
SADR	Description											
0	When read, indicates no register address/data received.											
1	When read, indicates address/data received.											
		Write 0 clear this bit and write 1 is ignored. When SADR is 1, it may trigger an interrupt depends on the interrupt enable setting.										
16	CADT	Command Address and Data Transmitted. This bit indicates that a CODEC register reading/writing command transmission has completed or not.	<table border="1"> <thead> <tr> <th>CADT</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>When read, indicates the command has not done.</td> </tr> <tr> <td>1</td> <td>When read, indicates the command has done.</td> </tr> </tbody> </table>		CADT	Description	0	When read, indicates the command has not done.	1	When read, indicates the command has done.	RW	
CADT	Description											
0	When read, indicates the command has not done.											
1	When read, indicates the command has done.											
		Write 0 clear this bit and write 1 is ignored. When CADT is 1, it may trigger an interrupt depends on the interrupt enable setting.										
15:0	Reserved	Writing has no effect, read as zero.			R							

12.2.8 AIC I2S/MSB-justified Status Register (I2SSR)

I2SSR is used to reflect AIC status in I2S/MSB-justified. It is a read-only register.

	I2SSR																0x1002001c															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								CHBSY	TBSY	RBSY	BSY	Reserved			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Writing has no effect, read as zero.	R

5	CHBSY	AIC Transmitter busy in I2S/MSB-justified format.(Multi-channel status)		R
		CHBSY	Description	
		0	AIC Transmitter part is idle or disabled.	
4	TBSY	AIC Transmitter busy in I2S/MSB-justified format.		R
		TBSY	Description	
		0	AIC Transmitter part is idle or disabled.	
3	RBSY	AIC Receiver busy in I2S/MSB-justified format.		R
		RBSY	Description	
		0	AIC Receiver part is idle or disabled.	
2	BSY	AIC busy in I2S/MSB-justified format.		R
		BSY	Description	
		0	AIC controller is idle or disabled.	
1	AIC controller currently is transmitting or receiving a frame.			
1:0	Reserved	Writing has no effect, read as zero.		R

12.2.9 AIC AC97 CODEC Command Address & Data Register (ACCAR, ACCDR)

ACCAR and ACCDR are used to hold register address and data for external AC-link CODEC register read/write operation through SDATA_OUT. The format of ACCAR.CAR and ACCDR.CDR is compliant with AC'97 Component Specification 2.3 where ACCAR.CAR[19] of "1" specifies CODEC register read operation, of "0" specifies CODEC register write operation. The write access to ACCAR and ACCDR signals AIC to issue this operation. Please reference to 12.4.4 for software flow. These registers are valid only in AC-link. It is ignored in I2S/MSB-justified format.

	ACCAR																														0x10020020	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												CAR																			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:20	Reserved	Writing has no effect, read as zero.	R
19:0	CAR	Command Address Register. This is used to hold 20-bit AC '97 CODEC register address transmitted in SDATA_OUT slot 1. After this field is write, it should not be write again until the operation is finished.	RW

ACCDR																0x10020024																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												CDR																			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:20	Reserved	Writing has no effect, read as zero.	R
19:0	CDR	Command Data Register. This is used to hold 20-bit AC'97 CODEC register data transmitted in SDATA_OUT slot 2. After this field is write, it should not be write again until the operation is finished.	RW

12.2.10 AIC AC97 CODEC Status Address & Data Register (ACSAR, ACSDR)

ACSAR and ACSDR are used to receive the external AC-link CODEC registers address and data from SDATA_IN. When AIC receives CODEC register status from SDATA_IN, it set ACSR.SADR bit and put the address and data to ACSAR.SAR and ACSDR.SDR. Please reference to 12.4.4 for software flow. These registers are valid only in AC-link format and are ignored in I2S/MSB-justified format.

ACSAR																0x10020028																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												SAR																			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:20	Reserved	Writing has no effect, read as zero.	R
19:0	SAR	CODEC Status Address Register. This is used to receive 20-bit AC '97 CODEC status address from SDATA_IN slot 1. Which reflect the register index for which data is being returned. The write operation is ignored.	R

ACSDR																0x1002002C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												SDR																			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:20	Reserved	Writing has no effect, read as zero.	R

19:0	SDR	CODEC Status Data Register. This is used to receive 20-bit AC '97 CODEC status data from SDATA_IN slot 2. The register data of external CODEC is returned. The write operation is ignored.	R
------	-----	--	---

12.2.11 AIC I2S/MSB-justified Clock Divider Register (I2SDIV)

I2SDIV is used to set clock divider to generated BIT_CLK from SYS_CLK in I2S/MSB-justified format.

I2SDIV																0x10020030																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved																IDV				Reserved				DV								
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	Name	Description	RW
31:12	Reserved	Writing has no effect, read as zero.	R
11:8	IDV	Audio IBIT_CLK clock divider value minus 1. I2SDIV.IDV is used to control the generating of IBIT_CLK from dividing SYS_CLK. The dividing value can be even or odd and I2SDIV.IDV should be set to the dividing value minus 1. IBIT_CLK frequency is fixed to $64 f_s$ in AIC, where f_s is the audio sample frequency. I2SDIV.IDV depends on SYS_CLK frequency f_{SYS_CLK} , which is selected according to external CODEC's requirement and internal PLL frequency. Please reference to 1.4.10 Serial Audio Clocks and Sampling Frequencies for further description.	RW
7:4	Reserved	Writing has no effect, read as zero.	R
3:0	DV	Audio BIT_CLK clock divider value minus 1. I2SDIV.DV is used to control the generating of BIT_CLK from dividing SYS_CLK. The dividing value can be even or odd and I2SDIV.IDV should be set to the dividing value minus 1. BIT_CLK frequency is fixed to $64 f_s$ in AIC, where f_s is the audio sample frequency. I2SDIV.DV depends on SYS_CLK frequency f_{SYS_CLK} , which is selected according to external CODEC's requirement and internal PLL frequency. Please reference to 1.4.10 Serial Audio Clocks and Sampling Frequencies for further description.	RW

12.2.12 AIC FIFO Data Port Register (AICDR)

AICDR is act as data input port to transmit FIFO when write and data output port from receive FIFO when read, one audio sample every time. The FIFO width is 24 bits. Audio sample with size N that is less than 24 is located in LSB N-bits. The sample size is specified by ACCR2.OASS and ACCR2.IASS in AC-link, and by I2SCR.WL in I2S/MSB-justified. The sample order is specified by ACCR1.XS and ACCR1.RS in AC-link. In I2S/MSB-justified, the left channel sample is prior to the right channel sample.

Care should be taken to monitor the status register to insure that there is room for data in the FIFO when executing a program read or write transaction. This is taken care automatically in DMA.

AICDR																0x10020034																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								DATA																							
RST	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:0	DATA	FIFO port. When write to it, data is push to the transmit FIFO. When read from it, data is pop from the receiving FIFO.	RW

12.2.13 SPDIF Enable Register (SPENA)

The register SPENA is used to trigger SPDIF transmitter to work.

SPENA																0x10020080							
Bit																7	6	5	4	3	2	1	0
	Reserved											Reserved					SPEN						
RST																0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:1	Reserved	Writing has no effect, read as zero.	R
0	SPEN	Enable / disable the SPDIF transmitter. 0: SPDIF transmitter is disabled 1: SPDIF transmitter is enabled	RW

12.2.14 SPDIF Control Register (SPCTRL)

The register SPCTRL is used to control SPDIF to work.

SPCTRL																0x10020084															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved											DMA_EN	D_TYPE	SIGN_N	INVALID	SFT_RST	SPDIF_I2S	Reserved					M_TRIG	M_FFUR							
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	Name	Description	RW								
15	DMA_EN	DMA transmitter enable bit. 0: DMA transmitter disable 1: DMA transmitter enable	RW								
14	D_TYPE	If the bit number of data is less than 16, the data in memory is as follows: 0: <table border="1" data-bbox="497 564 1315 656"> <tr> <td>XXXXXXXXXXXXXXXXXX</td> <td>Data 0</td> </tr> <tr> <td>XXXXXXXXXXXXXXXXXX</td> <td>Data 1</td> </tr> </table> 1: <table border="1" data-bbox="497 698 1315 790"> <tr> <td>Data 1</td> <td>Data 0</td> </tr> <tr> <td>Data 3</td> <td>Data 2</td> </tr> </table>	XXXXXXXXXXXXXXXXXX	Data 0	XXXXXXXXXXXXXXXXXX	Data 1	Data 1	Data 0	Data 3	Data 2	RW
XXXXXXXXXXXXXXXXXX	Data 0										
XXXXXXXXXXXXXXXXXX	Data 1										
Data 1	Data 0										
Data 3	Data 2										
13	SIGN_N	Signed to unsigned or not. If it is 1, the incoming and outgoing audio sample data will be transferred by toggle its most significant bit. 0: Not transfer 1: Do transfer	RW								
12	INVALID	Data invalid bit. The data transmitted on SPDIF is valid or not. 0: Valid 1: Invalid	RW								
11	SFT_RST	SPDIF FIFO software-reset. Set it to 1 and later it will be cleared by hardware auto. When SFT_RST returns back to 0, the FIFO finish reset. 0: Stop reset 1: Start reset	RW								
10	SPDIF_I2S	Choose SPDIF or I2S. 0: I2S 1: SPDIF									
9:2	Reserved	Writing has no effect, read as zero.	R								
1	M_TRIG	Trigger interrupt mask. 0: Enabled 1: Masked	RW								
0	M_FFUR	FIFO underrun interrupt mask. 0: Enabled 1: Masked	RW								

12.2.15 SPDIF State Register (SPSTATE)

The register SPSTATE is used to keep the state of SPDIF.

SPSTATE																0x10020088															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

																			Reserved	FIFO_LVL	BUSY	Reserved	F_TRIG	F_FFUR			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	Reserved	Writing has no effect, read as zero.	R
14:8	FIFO_LVL	FIFO level. The bits indicate the amount of valid data in FIFO.	R
7	BUSY	SPDIF busy bit. 0: SPDIF is not working 1: SPDIF is working	R
6:2	Reserved	Writing has no effect, read as zero.	R
1	F_TRIG	Trigger flag. 0: Not active 1: Active	R
0	F_FFUR	FIFO underrun flag. 0: Not active 1: Active	RW

12.2.16 SPDIF Configure 1 Register (SPCFG1)

The register SPCFG1 is used to configure SPDIF.

	SPCFG1																0x1002008C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved														INIT_LVL	ZRO_VLD	Reserved	TRIG	SRC_NUM	CH1_NUM	CH2_NUM												
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:18	Reserved	Writing has no effect, read as zero.	R
17	INIT_LVL	Initial level set bit. 0: SPDIF initial level is low 1: SPDIF initial level is high	
16	ZRO_VLD	The valid bit of channel state is 0 or 1 when play ZERO sample under FIFO underflow. 0: Valid 1: Invalid	RW
15:14	Reserved	Writing has no effect, read as zero.	R

13:12	TRIG	Specify the trigger value of FIFO.		RW
		TRIG	Description	
		00	Trigger Value is 4.	
		01	Trigger Value is 8.	
		10	Trigger Value is 16.	
		11	Trigger Value is 32.	
11:8	SRC_NUM	Source number. 0000:Unspecified 0001~1111:1~15		RW
7:4	CH1_NUM	Channel 1 number. 0000:Unspecified 0001~1111:A~O		RW
3:0	CH2_NUM	Channel 2 number. 0000:Unspecified 0001~1111:A~O		RW

12.2.17 SPDIF Configure 2 Register (SPCFG2)

The register SPCFG2 is used to configure SPDIF.

SPCFG2																0x10020090																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved		FS			ORG_FRQ			SAMPL_WL		MAX_WL		CLK_ACU		CAT_CODE						CH_MD		Reserved		PRE	COPY_N	AUDIO_N	CON_PRO				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:30	Reserved	Writing has no effect, read as zero.	R
29:26	FS	Sampling frequency. 0000:44.1kHz 0010:48kHz 0011:32kHz 1010:96kHz 1110:192kHz Others: Reference IEC60958-3	RW
25:22	ORG_FRQ	Original sampling frequency. 1111:44.1kHz 1101:48kHz 1100:32kHz 0101:96kHz 0001:192kHz	RW

		Others: Reference IEC60958-3	
21:19	SAMPL_WL	Sample word length. When MAX_WL=1: 001:20 bit 110:21 bit 010:22 bit 100:23 bit 101:24 bit Others: reserved When MAX_WL=0: 001:16 bit 110:17 bit 010:18 bit 100:19 bit 101:20 bit Others: reserved	RW
18	MAX_WL	Maximum audio sample word length. 0:20 bit; 1:24 bit.	RW
17:16	CLK_ACU	Clock Accuracy of transmitted clock. 00: Level II 01: Level I 10: Level III 11: Interface frame rate not matched to sampling frequency	RW
15:8	CAT_CODE	Category code. Reference IEC60958-3 for full details. 00 indicates "general" mode.	RW
7:6	CH_MD	Channel mode choose bit. 00: Mode 0 01~11: Reserved	RW
5:4	Reserved	Writing has no effect, read as zero.	R
3	PRE	Pre-emphasis set bit. 0: None 1: 15us/15us	RW
2	COPY_N	Copyright set bit. 0: Copyright is asserted 1: Copyright is not asserted	RW
1	AUDIO_N	Linear PCM identification bit. 0: Audio sample word represents linear PCM samples 1: Audio sample word used for other purpose	RW
0	CON_PRO	Consumer mode and professional mode choose bit. 0: Consumer mode 1: Professional mode Professional is not supported in the chip.	RW

12.2.18 SPDIF FIFO Register (SPFIFO)

The register SPCFG1 is used to configure SPDIF.

SPFIFO																0x10020094																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								DATA																							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:0	DATA	FIFO port. When write to it, data is push to the transmit FIFO. Read from it as 0.	W

12.3 Serial Interface Protocol

12.3.1 AC-link serial data format

Following figures are AC-link serial data format. Audio data is MSB adjusted, regardless of 8, 16, 18, 20, 24 bits sample size. When a 24-bits sample is transmitted, the LSB 4-bits are truncated. When try to record 24-bits sample, 4-bits of 0 are appended in LSB. Please reference to “AC '97 Component Specification Revision 2.3, 2002”, provided by Intel Corporation, for details of AC '97 architecture and AC-link specification.

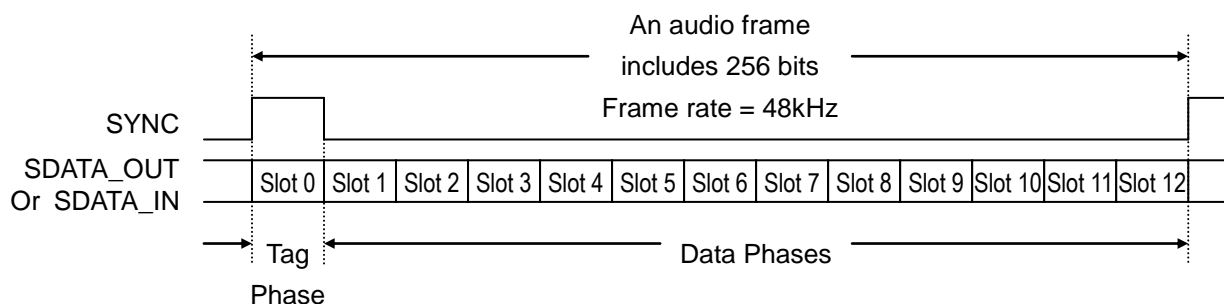


Figure 12-10 AC-link audio frame format

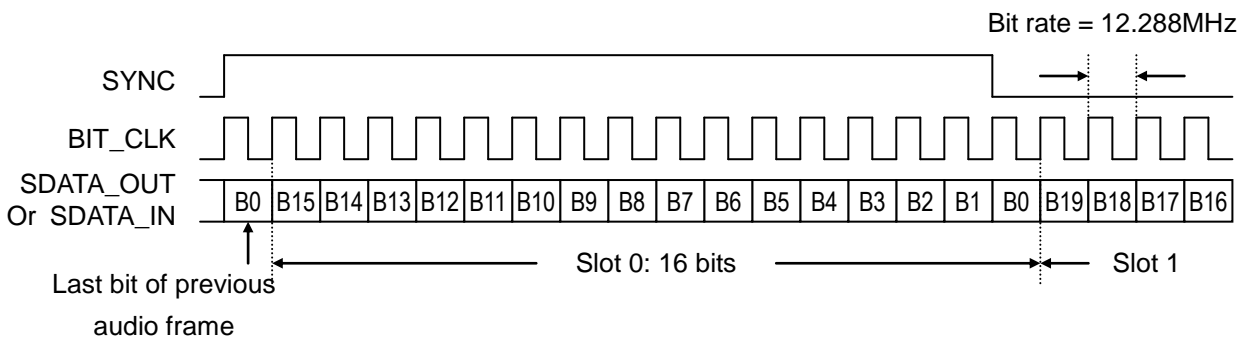


Figure 12-11 AC-link tag phase, slot 0 format

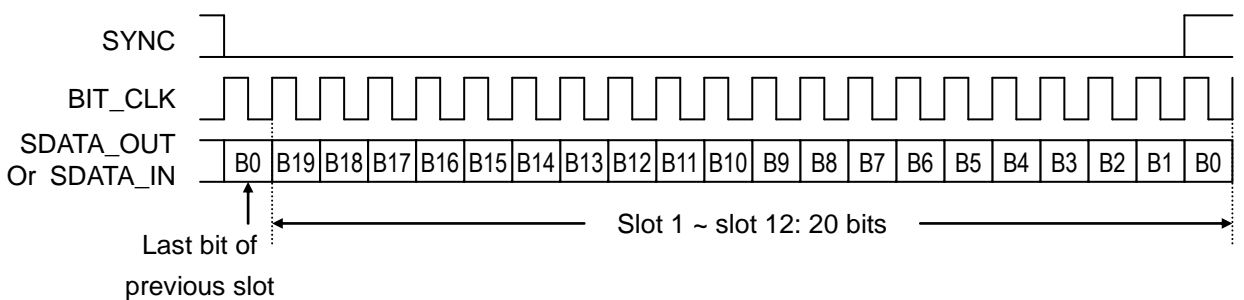


Figure 12-12 AC-link data phases, slot 1 ~ slot 12 format

12.3.2 I2S and MSB-justified serial audio format

Normal I2S and MSB-justified are similar protocols for digitized stereo audio transmitted over a serial path.

The BIT_CLK supplies the serial audio bit rate, the basis for the external CODEC bit-sampling logic. Its frequency is 64 times the audio sampling frequency. Divided by 64, the resulting 8 kHz to 48 kHz or even higher signal signifies timing for left and right serial data samples passing on the serial data paths. This left/right signal is sent to the CODEC on the SYNC pin. Each phase of the left/right signal is accompanied by one serial audio data sample on the data pins SDATA_IN and SDATA_OUT.

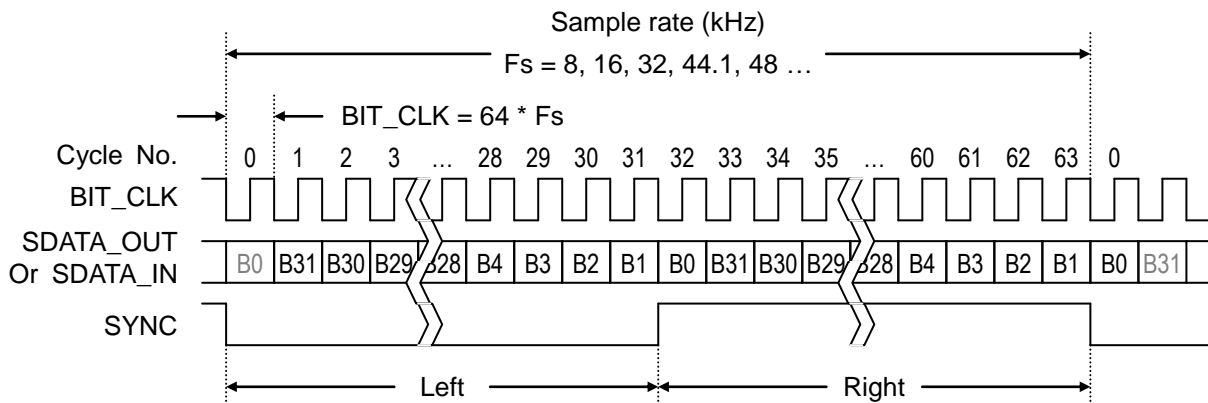


Figure 12-13 I2S data format (A: LR mode)

In the A: LR mode, first send the left channel in a stereo frame. One Left slot and one Right slot make a sample frame. It is the normal mode of I2S.

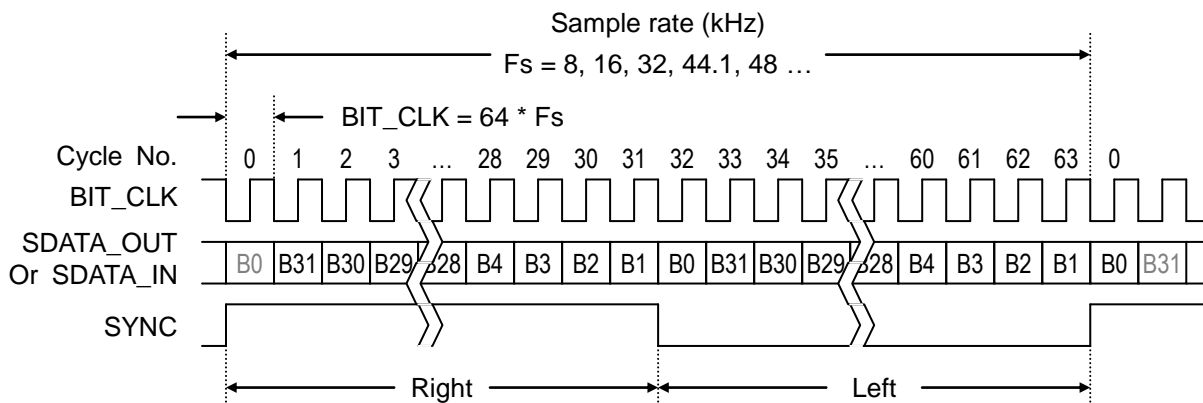


Figure 12-14 I2S data format (B: RL mode)

In the B: RL mode, first send the right channel in a stereo frame. One Right slot and one Left slot make a sample frame. It is used in same CODEC.

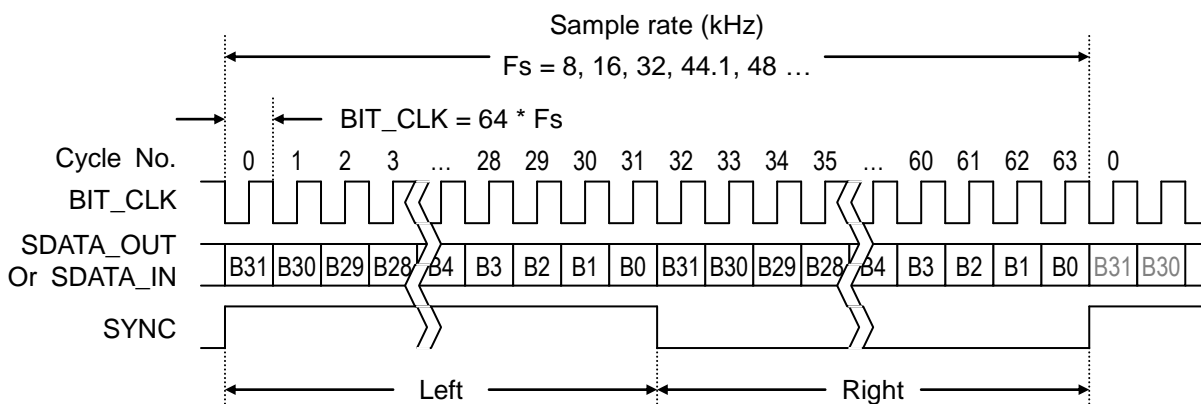


Figure 12-15 MSB-justified data format (C: LR mode)

In the C: LR mode, first send the left channel in a stereo frame. One Left slot and one Right slot make a sample frame. It is the normal mode in MSB-justified.

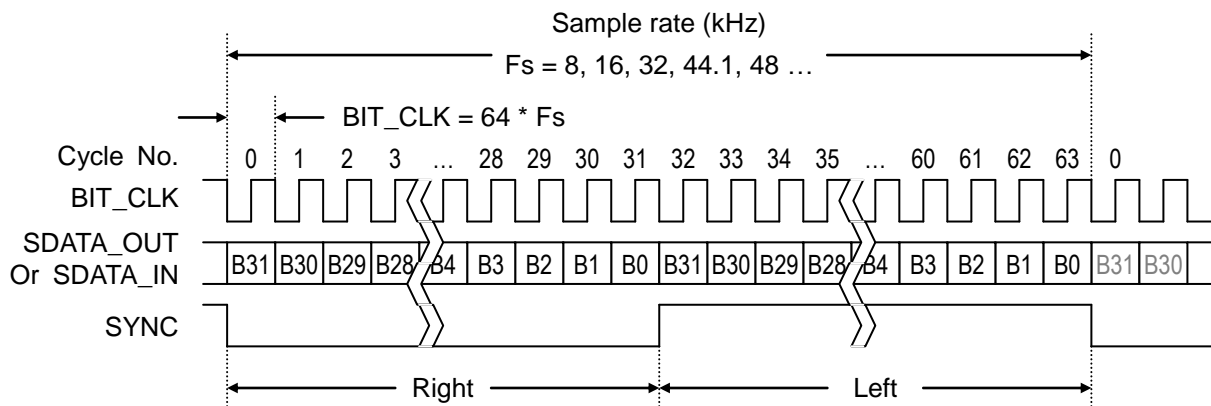


Figure 12-16 MSB-justified data format (D: RL mode)

In the D: RL mode, first send the right channel in a stereo frame. One Right slot and one Left slot make a sample frame.

Figure 12-13 and Figure 12-15 provide timing diagrams that show formats for the normal I2S and MSB-justified modes of operations. Data is sampled on the rising edge of the BIT_CLK and data is sent out on the falling edge of the BIT_CLK.

Data is transmitted and received in frames of 64 BIT_CLK cycles (If BIT_CLK is generated internally). Each frame consists of a left sample and a right sample. Each sample holds 8, 16, 18, 20 or 24 bits of valid data. The LSB other bits of each sample is padded with zeroes.

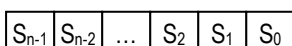
In the normal I2S mode, the SYNC is low for the left sample and high for the right sample. Also, the MSB of each data sample lags behind the SYNC edges by one BIT_CLK cycle.

In the MSB-justified mode, the SYNC is high for the left sample and low for the right sample. Also, the MSB of each data sample is aligned with the SYNC edges.

When use with the internal CODEC, the BIT_CLK and SYNC signals also with O_BIT_CLK and O_SYNC signals are provided by the internal CODEC from the SYSCLK, which is enabled by I2SCR.ESCLK and configured to 12MHz clock using CPM.

12.3.3 Audio sample data placement in SDATA_IN/SDATA_OUT

The placement of audio sample in incoming/outgoing serial data stream for all formats support in AIC is MSB (Most Significant Bit) justified. Suppose n bit sample composed by



Following table described the how sample data bits are transferred.

Table 12-4 Sample data bit relate to SDATA_IN/SDATA_OUT bit

AC-link Format						I2S/MSB-Justified Format					
SDATA IN/OUT	Audio Sample Size (bit)					SDATA IN/OUT	8	16	18	20	24
	8	16	18	20	24						
B19	S7	S15	S17	S19	S23	B31	S7	S15	S17	S19	S23
B18	S6	S14	S16	S18	S22	B30	S6	S14	S16	S18	S22
B17	S5	S13	S15	S17	S21	B29	S5	S13	S15	S17	S21
B16	S4	S12	S14	S16	S20	B28	S4	S12	S14	S16	S20
B15	S3	S11	S13	S15	S19	B27	S3	S11	S13	S15	S19
B14	S2	S10	S12	S14	S18	B26	S2	S10	S12	S14	S18
B13	S1	S9	S11	S13	S17	B25	S1	S9	S11	S13	S17
B12	S0	S8	S10	S12	S16	B24	S0	S8	S10	S12	S16
B11	0	S7	S9	S11	S15	B23	0	S7	S9	S11	S15
B10	0	S6	S8	S10	S14	B22	0	S6	S8	S10	S14
B9	0	S5	S7	S9	S13	B21	0	S5	S7	S9	S13
B8	0	S4	S6	S8	S12	B20	0	S4	S6	S8	S12
B7	0	S3	S5	S7	S11	B19	0	S3	S5	S7	S11
B6	0	S2	S4	S6	S10	B18	0	S2	S4	S6	S10
B5	0	S1	S3	S5	S9	B17	0	S1	S3	S5	S9
B4	0	S0	S2	S4	S8	B16	0	S0	S2	S4	S8
B3	0	0	S1	S3	S7	B15	0	0	S1	S3	S7
B2	0	0	S0	S2	S6	B14	0	0	S0	S2	S6
B1	0	0	0	S1	S5	B13	0	0	0	S1	S5
B0	0	0	0	S0	S4	B12	0	0	0	S0	S4
						B11	0	0	0	0	S3
						B10	0	0	0	0	S2
						B9	0	0	0	0	S1
						B8	0	0	0	0	S0
						B7~ B0	0	0	0	0	0

If in 16 bits packed mode, the data transferred is the same as the 16 bits normal mode as shown above. But there are two samples in one word.

12.3.4 SPDIF Protocol

SPDIF block format is shown below:

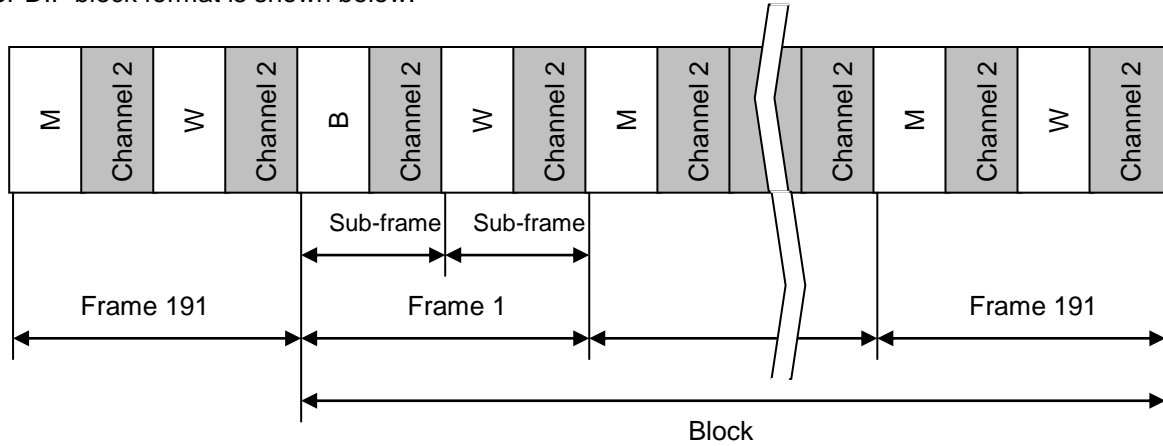


Figure 12-17 Block format

Sub-frame format in PCM mode is shown below:

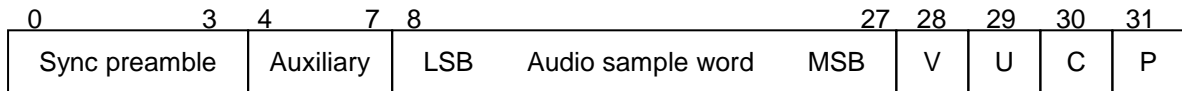


Figure 12-18 Sub-frame format in PCM mode

Sub-frame format in non-PCM mode is shown below:

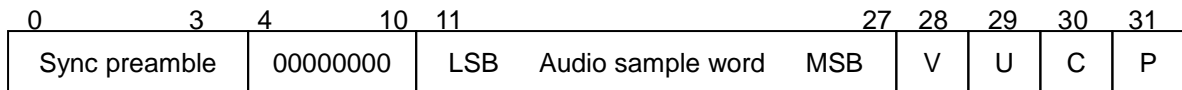


Figure 12-19 Sub-frame format in non-PCM mode

12.4 AC97/I2S Operation

The AIC can be accessed either by the processor using programmed I/O instructions or by the DMA controller. The processor uses programmed I/O instructions to access the AIC and can access the following types of data.

- **The AIC memory mapped registers data**—All registers are 32 bits wide and are aligned to word boundaries.
- **AIC controller FIFO data**—An entry is placed into the transmit FIFO by writing to the I2S controller's Serial Audio Data register (AICDR). Writing to AICDR updates a transmit FIFO entry. Reading AICDR flushes out a receive FIFO entry.

- **The external CODEC registers for I2S CODEC**—CODEC registers can be accessed through the L3 bus. The L3 bus operation is emulated by software controlling three GPIO pins.
- **The external CODEC registers for AC97 CODEC**—An AC97 audio CODEC can contain up to sixty-four 16-bit registers. A CODEC uses a 16-bit address boundary for registers. The AIC supplies access to the CODEC registers through several registers.
- **The internal CODEC registers** can be accessed via memory-mapped registers in the CODEC.

The DMA controller can only access the FIFOs. Accesses are made through the data registers, as explained in the previous paragraph. The DMA controller responds to the following DMA requests made by the I2S controller:

- The transmit FIFO request is based on the transmit trigger-threshold (AICFR.TFTH) setting. See 12.2.1 for further details regarding AICFR.TFTH.
- The receive FIFO request is based on the receive trigger-threshold (AICFR.RFTH) setting. See 12.2.1 for further details regarding AICFR.RFTH.

Before operation to AIC, you may need to set proper PIN function selection from GPIO using if the pin is shared with GPIO.

Please also reference to “AC '97 Component Specification Revision 2.3, 2002” when deal with AIC AC-link operations.

12.4.1 Initialization

At power-on or other hardware reset (WDT and etc), AIC is disabled. Software must initiate AIC and the internal or external CODEC after power-on or reset. If errors found in data transferring, or in other places, software must initial AIC and optional, the internal or external CODEC. Here is the initial flow.

- 1 Select internal or external CODEC (AICFR.ICDC).
- 2 If external CODEC is selected, select AC-link or I2S/MSB-Justified (AICFR.AUSEL). If internal CODEC is used, select I2S/MSB-Justified format (AICFR.AUSEL=1). If the resettlement without involving link format and architecture changing, this step can be skip.
- 3 If I2S/MSB-Justified is selected, select between I2S and MSB-Justified (I2SCR.AMSL).
- 4 Decide BIT_CLK direction (AICFR.BCKD) and SYNC direction (AICFR.SYNCD).
- 5 If BIT_CLK is configured as output, BIT_CLK divider I2SDIV.DV must be set to what correspond with the values as shown in Table 12-8. And the clock selection and the divider between PLL clock out and AIC also must be set (CFR.I2S and I2SCDR in CPM). If internal CODEC is used, select 12MHz clock input (via set proper value in CFR.I2S and I2SCDR), I2S format (I2SCR.AMSL=0), input BIT_CLK (AICFR.BCKD=0), input SYNC (AICFR.SYNCD=0).
- 6 Enable AIC by write 1 to AICFR.ENB.
- 7 If it needs to reset AIC registers and flush FIFOs, write 1 to AICFR.RST. If need only flush

- FIFOs, write 1 to AICCR.FLUSH. BIT_CLK must exist here and after.
- 8 In AC-link format, issue a warm or cold CODEC reset.
 - 9 In AC-link format, configure AC '97 CODEC via ACCAR and ACCDR registers. If the resettlement doesn't involving AC'97 CODEC registers changing, this step can be skipped.
 - 10 In case of external CODEC with I2S/MSB-Justified format, configure I2S/MSB-justified CODEC via the control bus connected to the CODEC, for instance SMB or L3, depends on CODEC. In case of internal CODEC, configure CODEC via CODEC's memory mapped registers. If the resettlement without involving I2S/MSB-justified CODEC or ADC/DAC function changing, this step can be skip.

12.4.2 AC '97 CODEC Power Down

AC '97 CODEC can be placed in a low power mode. When the CODEC's power-down register (26h), is programmed to the appropriate value, the CODEC will be put in a low power mode and both BIT_CLK and SDATA_IN will be brought to and held at a logic low voltage level.

Once powered down, re-activation of the AC-link via re-assertion of the SYNC signal must not occur for a minimum of four audio frame times following the frame in which the power down was triggered. When AC-link powers up it indicates readiness via the CODEC Ready bit (input slot 0, bit 15).

12.4.3 Cold and Warm AC '97 CODEC Reset

AC-link reset operations occur when the system is initially powered up, when resuming from a lower powered sleep state, and in response to critical subsystem failures that can only be recovered from with a reset.

12.4.3.1 Cold AC '97 CODEC Reset

A cold reset is achieved by asserting RESET# for the minimum specified time. By driving RESET# low, BIT_CLK, and SDATA_IN will be activated, or re-activated as the case may be, and all AC '97 CODEC registers will be initialized to their default power on reset values.

RESET# is an asynchronous AC '97 CODEC input.

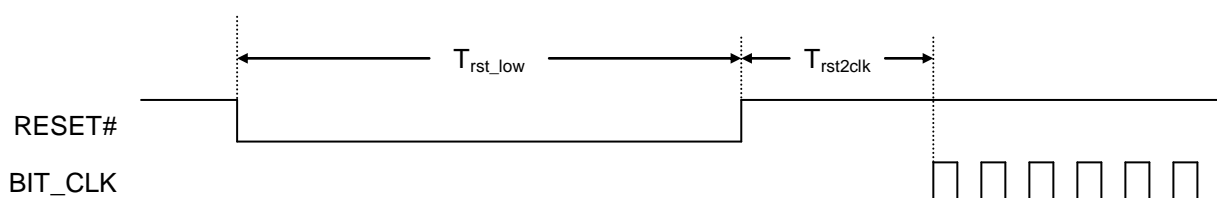


Figure 12-20 Cold AC '97 CODEC Reset Timing

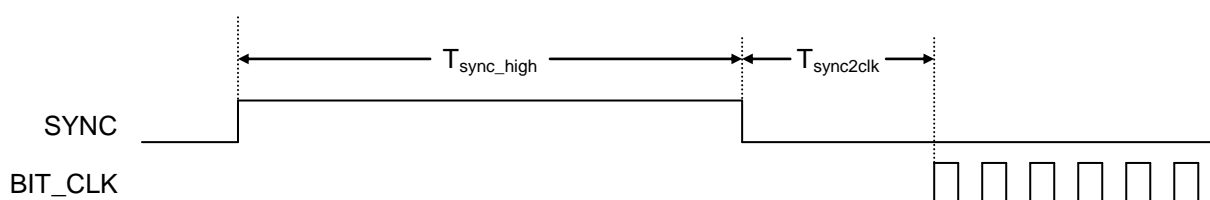
Table 12-5 Cold AC '97 CODEC Reset Timing parameters

Parameter	Symbol	Min	Type	Max	Units
RESET# active low pulse width	T_{rst_low}	1.0	-	-	μ s
RESET# inactive to BIT_CLK startup delay	$T_{rst2clk}$	162.8	-	-	ns

12.4.3.2 Warm AC '97 CODEC Reset

A warm AC'97 reset will re-activate the AC-link without altering the current AC'97 register values. Driving SYNC high for a minimum of 1 μ s in the absence of BIT_CLK signals a warm reset.

Within normal audio frames SYNC is a synchronous AC '97 CODEC input. However, in the absence of BIT_CLK, SYNC is treated as an asynchronous input used in the generation of a warm reset to AC '97 CODEC.

**Figure 12-21 Warm AC '97 CODEC Reset Timing****Table 12-6 Warm AC '97 CODEC Reset Timing Parameters**

Parameter	Symbol	Min	Type	Max	Units
SYNC active high pulse width	T_{sync_high}	1.0	-	-	Ms
SYNC inactive to BIT_CLK startup delay	$T_{sync2clk}$	162.8	-	-	Ns

12.4.4 External CODEC Registers Access Operation

The external audio CODEC can be configured/controlled by its internal registers. To access these registers, an I2S/MSB-justified CODEC usually employs L3 bus, SPI bus, SMB bus or other control bus. The L3 bus operation can be emulated by software by using 3 GPIO pins of the chip. For AC '97, "AC '97 Component Specification" defines the CODEC register access protocol. Several registers are provided in AIC to accomplish this task.

The ACCAR and ACCDR are used to send a register accessing request command to external AC'97 CODEC. The ACSAR and ACSDR are used to receive a register's content from external AC'97 CODEC. The register accessing request and the register's content returning is asynchronous.

The AC'97 CODEC register accessing request flow:

- 1 If ACSR.CADT is 0, wait for 25.4 μ s. If no previous accessing request, this step can be skip.
- 2 Clear ACSR.CADT.
- 3 If read access, write read-command and register address to ACCAR, if write access, write write-command and register address to ACCAR and write data to ACCDR. Any order of write ACCAR and ACCDR is OK.
- 4 Polling for ACSR.CADT changing to 1, which means the request has been send to CODEC via AC-link.

The AC'97 CODEC register content receiving flow by polling:

- 1 Polling for ACSR.SADR changing to 1.
- 2 Read the CODEC register's address from ACSAR and content from ACSDR.
- 3 Clear ACSR.SADR.

The AC'97 CODEC register content receiving flow by interrupt:

- 1 Before accessing request, clear ACSR.SADR and set ACCR2.ESADR.
- 2 Waiting for the interrupt. When the interrupt is found, check if ACSR.SADR is 1, if not, repeat this step again.
- 3 Read the CODEC register's address from ACSAR and content from ACSDR.
- 4 Clear ACSR.SADR.

12.4.5 Audio Replay

Outgoing audio sample data (from AIC to CODEC) is written to AIC transmit FIFO from processor via store instruction or from memory via DMA. AIC then takes the data from the FIFO, serializes it, and sends it over the serial wire SDATA_OUT to an external CODEC or over an internal wire to an internal CODEC.

The audio transmission is enabled automatically when the AIC is enabled by set AICFR.ENB. But all replay data is zero at this time except both of the following conditions are true:

- 1 AICCR.ERPL must be 1. If AICCR.ERPL is 0, value of zero is send to CODEC even if there are samples in transmit FIFO.
- 2 At least one audio sample data in the transmit FIFO. If the transmit FIFO is empty, value of zero or last sample depends on AICFR.LSMP, is send to CODEC even if AICCR.ERPL is 1.

Here is the audio replay flow:

- 1 Configure the CODEC as needed.
- 2 Configure sample size by AICCR.OSS.
- 3 Configure sample channels (AICCR.CHANNEL).
- 4 If sample size is configured 16 bit, select packed or unpacked mode (AICCR.PACK16).
- 5 If two channels is configured, select the right-channel-first sample data or not (I2SCR.RFIRST).
- 6 If two channels is configured, select the sample data switched or not (I2SCR.SWLH).

- 7 Configure sample rate by clock dividers (for I2S/MSB-Justified format with BIT_CLK is provided internally) or by CODEC registers (for AC-link or BIT_CLK provided by external CODEC) or by accessing CODEC internal registers (for internal CODEC).
- 8 For AC-link, configure replay channels by ACCR1.XS.
- 9 Some other configurations: mono to stereo, endian switch, signed/unsigned data transfer, transmit FIFO configuration, play ZERO or last sample when TX FIFO under-run, and etc.
- 10 Write 1 to AICCR.ERPL.
It is suggested that at least a frame of PCM data is pre-filled in the transmit FIFO to prevent FIFO under-run flag (AICSR.TUR).
But when using internal CODEC, write first frame of PCM data to transmit FIFO till TX FIFO under-run (AICSR.TUR is set to 1), otherwise left/right channel may be switched.
- 11 Fill sample data to the transmit FIFO. Repeat this till finish all sample data. In this procedure, please control the FIFO to make sure no FIFO under-run and other errors happen. When the transmit FIFO under-run, noise or pause may be heard in the audio replay, AICSR.TUR is 1, and if AICCR.ETUR is 1, AIC issues an interrupt. Please reference to 12.4.7 for detail description on FIFO.
- 12 Waiting for AICSR.TFL change to 0. So that all samples in the transmit FIFO has been replayed, then we can have a clean start up next time.
- 13 Write 0 to AICCR.ERPL.

NOTE: Before replaying Open ADC BITCLK and close it to generating Record internal circuit reset when using internal CODEC.

12.4.6 Audio Record

Incoming audio sample data (from CODEC to AIC) is received from SDATA_IN (for an external CODEC) or an internal wire (for an internal CODEC) serially and converted to parallel word and stored in AIC receive FIFO. Then the data can be taken from the FIFO to processor via load instruction or to memory via DMA.

The audio recording is enabled automatically when the AIC is enabled by set AICFR.ENB. But all received data is discarded at this time except both of the following conditions are true:

- 1 AICCR.EREC must be 1. If AICCR.EREC is 0, the received data is discarded even if there are rooms in the receive FIFO.
- 2 At least one room left in the receive FIFO. If the receive FIFO is full, the received data is discarded even if AICCR.EREC is 1.

Here is the audio record flow:

- 1 Configure the CODEC as needed.
- 2 Configure sample size by AICCR.ISS.
- 3 Configure sample rate by clock dividers (for I2S/MSB-Justified format with BIT_CLK is provided internally) or by CODEC registers (for AC-link or BIT_CLK provided by external CODEC) or by CODEC memory mapped registers (for internal CODEC).

- 4 Some other configurations: signed/unsigned data transfer, receive FIFO configuration, and etc.
- 5 Write 1 to AICCR.EREC. Make sure there are rooms available in the receive FIFO before set AICCR.EREC. Usually, it should empty the receive FIFO by fetch data from it before set AICCR.EREC.
- 6 Take sample data form the receive FIFO. Repeat this till the audio finished. In this procedure, please control the FIFO to make sure no FIFO over-run and other errors happen. When the receive FIFO over-run, same recorded audio samples will be lost, AICSR.ROR is 1, and if AICCR.EROR is 1, AIC issues an interrupt. Please reference to 12.4.7 for detail description on FIFO. For AC-link, ACCR1.RS tells which channels are recorded. When using internal CODEC, the first data should be ignored.
- 7 Write 0 to AICCR.EREC.
- 8 Take sample data from the receive FIFO until AICSR.RFL change to 0. So that all samples in the receive FIFO has been taken away, then we can have a clean start up next time. When the receive FIFO is empty, read from it returns zero.

12.4.7 FIFOs operation

AIC has two FIFOs, one for transmit audio sample and one for receive. All AIC played/recorded audio sample data is taken from/send to transmit/receive FIFOs. The RX FIFO is in 24 bits width and 32 entries depth, one entry for keeping one audio sample regardless of the sample size. The TX FIFO is in 32 bits width and 64 entries depth, one entry for keeping one audio sample regardless of the sample size, but in 16 bits packed mode, one entry for keeping two audio samples. AICDR.DATA provides the access point for processor/DMA to write to transmit FIFO and read from receive FIFO. One time access to AICDR.DATA process one sample. The sample data should be put in LSB (Least Significant Bit) in memory or processor registers. For transmitting, bits exceed sample are discarded. For receiving, these bits are set to 0. Figure 12-22 illustrates the FIFOs access.

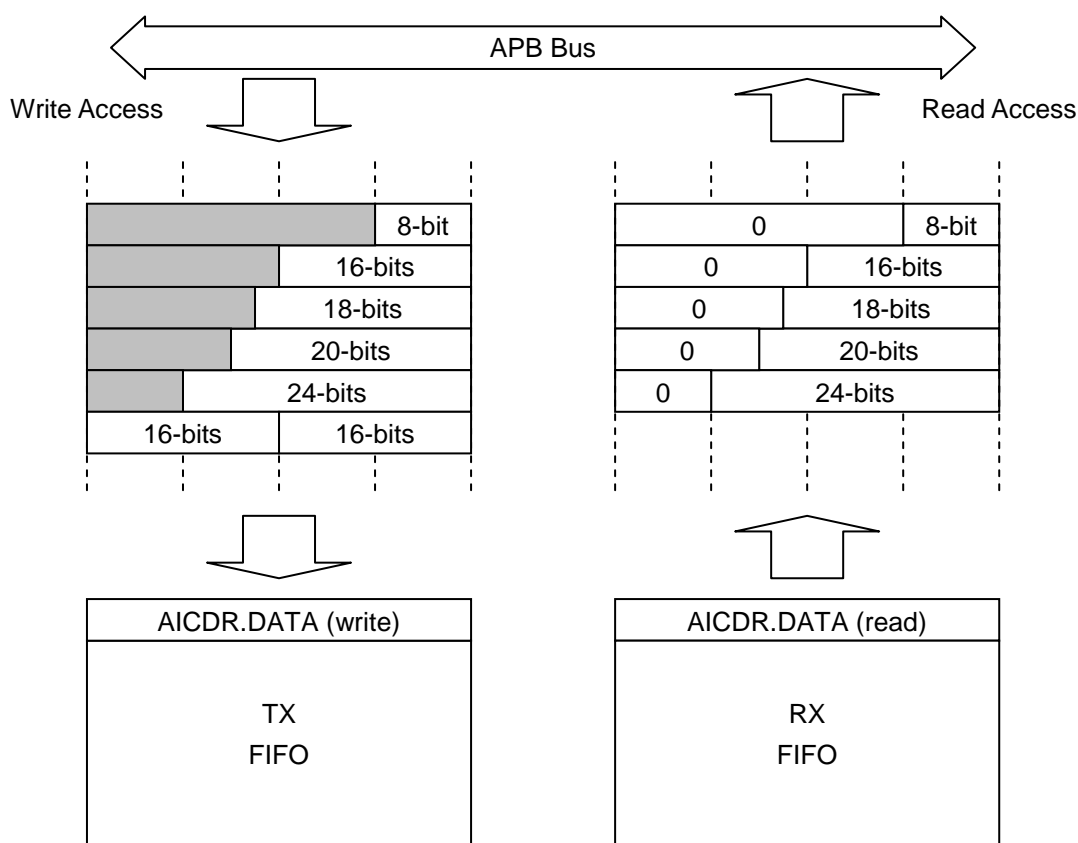


Figure 12-22 Transmitting/Receiving FIFO access via APB Bus

The software and bus initiator must guarantee the right sample placement at the bus.

In case of DMA bus initiator, one 24, 20, 18 bits audio sample must occupies one 32-bits word in memory, so 32-bits width DMA must be used. One 16 bits sample occupies one 16-bits half word in memory, so 16-bits width DMA must be used. One 8-bits sample occupies one byte in memory, and use 8-bits width DMA except 16bits packed mode. If in 16 bits packed mode, Two 16 bits sample occupies one 32-bits word in memory, so 32-bits width DMA must be used.

In case of processor bus initiator, any type of the audio sample must occupy one CPU general-purpose register at LSB, and read/write from/to AICDR.DATA with 32-bits load/store instruction. When process small sample size, 16-bits or 8-bits, software may need to do the data pack/unpack except 16 bits packed mode. In the 16bits packed mode, the sample data is packed, and two 16 bits audio samples occupy one CPU general-purpose register.

The AICFR.TFTH and AICFR.RFTH are used to set the FIFO level thresholds, which are the trig levels of DMA request and/or FIFO service interrupt. The AICFR.TFTH and AICFR.RFTH should be set to proper value; too small or too big are not good. When AICFR.RFTH is too small, or AICFR.TFTH is too big, the DMA burst length or the number of sample can be processed by processor is too small, which

harms the bus or processor efficiency. When AICFR.RFTH is too big or AICFR.TFTH is too small, the bus or the interrupt latency left for under-run/over-run is too small, which may causes replay/record errors.

AICSR.TUR is set to 1 during transmit under-run conditions. If AICCR.ETUR is 1, this can trigger an interrupt. During transmit under-run conditions, zero or last sample is continuously sent out across the serial link. Transmit under-run can occur under the following conditions:

- 1 Valid transmit data is still available in memory, but the DMA controller/processor starves the transmit FIFO, as it is busy servicing other higher-priority tasks.
- 2 The DMA controller/processor has transferred all valid data from memory to the transmit FIFO.

AICSR.ROR is set to 1 during receive over-run conditions. If AICCR.EROR is 1, this can trigger an interrupt. During receive over-run conditions, data sent by the CODEC is lost and is not recorded.

When replay/record two channels data, the left channel is default the first data in FIFOs and in the serial link. If multiple channels in AC-link are used, the channel sample order is follows the slot order. In 16bits packed mode, could configure that the left channel is the first data or the right channel. By default, the 16 bits LSB is left channel, 16 bits MSB is the right channel. But it also could be switched the Left or the Right channel (I2SCR.SWLH).

12.4.8 Data Flow Control

There are three approaches provided to control/synchronize the audio incoming/outgoing data flow.

12.4.8.1 Polling and Processor Access

AICSR.RFL and AICSR.TFL reflect how many samples exist in receiving and transmitting FIFOs. Through read these register fields, processor can detect when there are samples in receiving FIFO in audio record and then load them from the RX-FIFO, and when there are rooms in transmitting FIFO in audio replay and then store samples to the TX-FIFO.

Polling approach is in very low efficiency and is not recommended.

12.4.8.2 Interrupt and Processor Access

Set proper values to AICFR.TFTH and AICFR.RFTH, the FIFO interrupts trig thresholds. Set AICCR.ETFS and/or AICCR.ERFS to 1 to enable transmitting and/or receiving FIFO level trigger interrupts. When the interrupt found, it means there are rooms or samples in the TX or RX FIFO, and processor can store or load samples to or from the FIFO.

Interrupt approach is more efficient than polling approach.

12.4.8.3 DMA Access

Audio data is real time stream, though it is in low data bandwidth, usually less than 1.2Mbps. DMA approach is the most efficient and is the recommended approach.

To enable DMA operation, set AICCR.TDMS and AICCR.RDMS to 1 for transmit and receive respectively. It also needs to allocate two channels in DMA controller for data transmitting and receiving respectively. Please reference to the processor's DMA controller spec for the details.

The AICFR.TFTH and AICFR.RFTH are used to set the transmitting and receiving FIFO level thresholds, which determine the issuing of DMA request to DMA controller. To respond the request, DMAC initiator and controls the data movement between memory and TX/RX FIFO.

12.4.9 Audio Samples format

12.4.9.1 16 bits packed mode

One channel (mono) mode and two channels (stereo) mode:

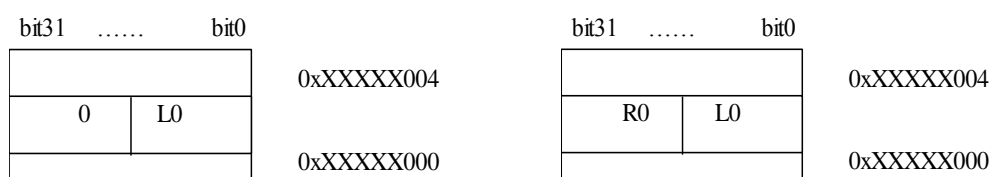


Figure 12-23 One channel (Left) and Two channels (right) mode (16 bits packed mode)

Four channels mode and six channels mode:

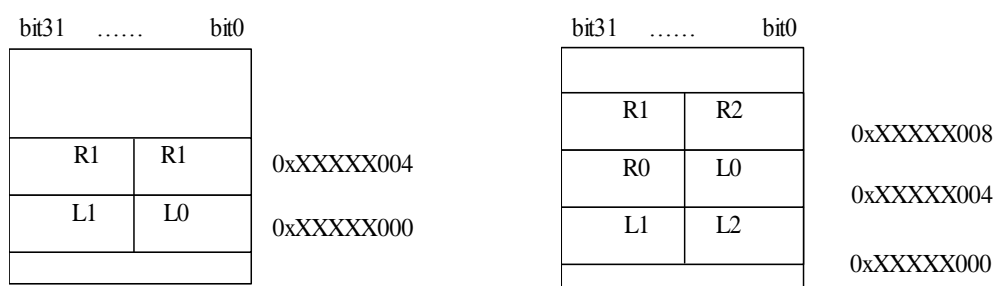


Figure 12-24 Four channels (Left) and Six channels (right) mode (16 bits packed mode)

Eight channels mode:

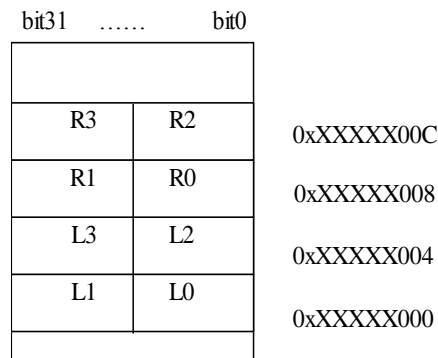


Figure 12-25 Eight channels mode (16 bits packed mode)

12.4.9.2 Normal mode.

One channel (Mono) and two channels (stereo) mode:

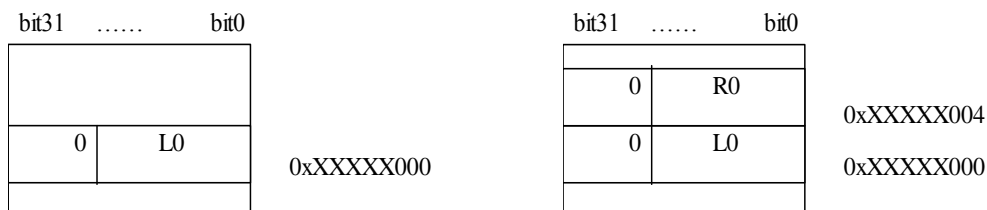


Figure 12-26 One channel (Left) and Two channels (right) mode

Four channels mode and six channels mode:

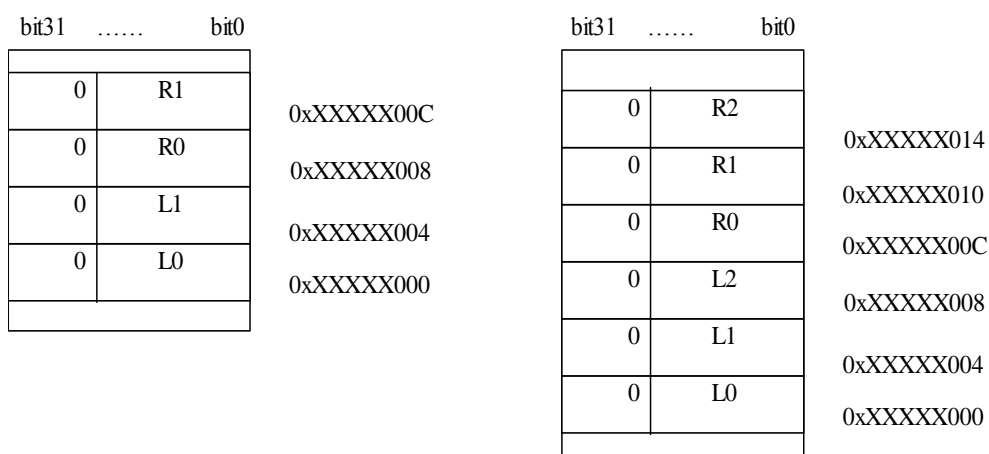


Figure 12-27 Four channels (Left) and Six channels (right) mode

Eight channel mode:

bit31 bit0		
0	R3	0xFFFFFFFF01C
0	R2	0xFFFFFFFF018
0	R1	0xFFFFFFFF014
0	R0	0xFFFFFFFF010
0	L3	0xFFFFFFFF00C
0	L2	0xFFFFFFFF008
0	L1	0xFFFFFFFF004
0	L0	0xFFFFFFFF000

Figure 12-28 Eight channels mode

12.4.10 Serial Audio Clocks and Sampling Frequencies

For internal CODEC, CODEC module containing the audio CODEC circuit/logic and corresponding controlling registers. CODEC needs a 12MHz clock from CPM called SYS_CLK and provides I_BITCLK, O_BITCLK and I_SYNC, O_SYNC (left-right clock which is the sample rate as ADC or DAC) to AIC for outgoing and incoming audio respectively. These clocks change when change the sample rate in CODEC controlling registers. When using internal CODEC, must configure SYNC and BIT_CLK as input, more details refers to [CODEC Spec](#).

For AC-link, the bit clock is input from chip external and is fixed to 12.288MHz. The sample frequency of 48kHz is supported in nature. Variable Sample Rate feature is supported in this AIC. If the CODEC supports this feature, sample rate other than 48kHz audio data can be replay directly. Otherwise, software has to do the rate transfer to replay other sample rate audio data. Double rate, 96kHz or even 88.2kHz audio is also supported with proper CODEC.

Following are for BIT_CLK/SYS_CLK configuration in I2S/MSB-Justified format with external CODEC.

The BIT_CLK is the rate at which audio data bits enter or leave the AIC. BIT_CLK can be supplied either by the CODEC or an internally PLL. If it is supplied internally, BIT_CLK is configured as output pins, and is supplied out to the CODEC. If BIT_CLK is supplied by the CODEC, then it is configured as an input pin. Register bit AICFR.BCKD is used to select BIT_CLK direction.

The audio sampling frequency is the frequency of the SYNC signal, which must be 1/64 of BIT_CLK, $f_{\text{BIT_CLK}} = 64 f_s$. But SYNC signal frequency is not fixed when using internal CODEC.

SYS_CLK is only for CODEC. It usually takes one of the two roles, as CODEC master clock input or as

CODEC over-sampling clock input. If SYS_CLK roles as CODEC master clock input, it usually should be set to a fixed frequency according to CODEC requirement but independent to audio sample rate. In this case, usually there is a PLL in the CODEC and CODEC roles master mode. See Figure 12-3 for the interface diagram. This is the recommended AIC CODEC system configuration.

If SYS_CLK roles as CODEC over-sampling clock, its frequency is usually 4, 6, 8 or 12 times of BIT_CLK frequency, which are 256, 384, 512 and 768 times of audio sample rates. Following table lists the relation between sample rate, BIT_CLK and SYS_CLK frequencies.

Table 12-7 Audio Sampling rate, BIT_CLK and SYS_CLK frequencies

Sample Rate f_s (kHz)	BIT_CLK (MHz) $f_{BIT_CLK} = 64 f_s$	SYS_CLK (MHz)			
		$256 f_s$	$384 f_s$	$512 f_s$	$768 f_s$
48	3.072	12.288	18.432	24.576	36.864
44.1	2.8224	11.2896	16.9344	22.5792	33.8688
32	2.048	8.192	12.288	16.384	24.576
24	1.536	6.144	9.216	12.288	18.432
22.05	1.4112	5.6448	8.4672	11.2896	16.9344
16	1.024	4.096	6.144	8.192	12.288
11.025	0.7056	2.8224	4.2336	5.6448	8.4672
8	0.512	2.048	3.072	4.096	6.144

In this processor, SYS_CLK can be selected from EXCLK or generated by dividing the PLL output clock in a CPM divider controlled by I2SCDR. If BIT_CLK is chosen as an output, another divider in AIC is used to divide SYS_CLK for it.

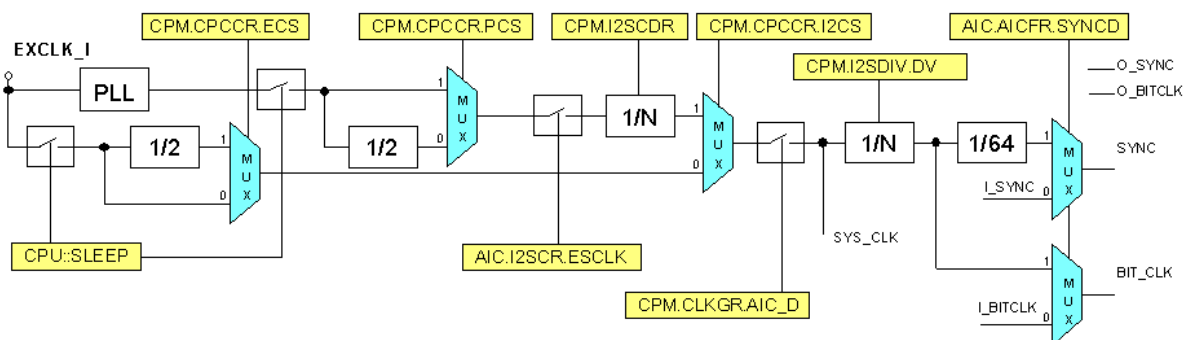


Figure 12-29 SYS_CLK, BIT_CLK and SYNC generation scheme

The setting of I2SDIV.DV is shown in following table.

Table 12-8 BIT_CLK divider setting

I2SDIV.DV	$f_{\text{SYS_CLK}}$	$f_{\text{BIT_CLK}}$	$f_{\text{SYS_CLK}} / f_{\text{BIT_CLK}}$
0x1	128 f_s	64 f_s	2
0x2	196 f_s	64 f_s	3
0x3	256 f_s	64 f_s	4
0x5	384 f_s	64 f_s	6
0x7	512 f_s	64 f_s	8
0xB	768 f_s	64 f_s	12

As we observe in Table 12-7, if SYS_CLK is taken as over-sampling clock by CODEC, the common multiple of all SYS_CLK frequencies is much bigger than the PLL output clock frequency. To generate all different SYS_CLK frequencies, one approach is change PLL frequency according to sample rate. This is not realistic, since frequently change PLL frequency during normal operation is not recommended.

Another approach is to found some approximate common multiples of all SYS_CLK frequencies according to the fact that there tolerance in audio sample rate. Take $f_{\text{SYS_CLK}} = 256 f_s$, Table 12-9 list most frequencies, which are less than 400MHz, with relatively small sample rate errors. It is suggested to set PLL frequency as close to the frequencies listed as possible, then use clock dividers to generate different SYS_CLK/BIT_CLK for different sample rate.

Table 12-9 Approximate common multiple of SYS_CLK for all sample rates

Approximate Common Frequency (MHz)	Max Error Caused in Audio Sample Rate (%)
123.53	0.53
147.11	0.24
170.68	0.79
235.5	0.87
247.06	0.53
270.64	0.11
280.56	0.73
294.22	0.24
305.14	0.67
317.79	0.53
329.57	0.66
341.35	0.79
347	0.85
353.13	0.90
358.79	0.69

370.59	0.53
382.96	0.54
394.17	0.24

Take PLL = 270.64 MHz as an example, following table lists the divider settings for various sample rates.

Table 12-10 CPM/AIC clock divider setting for various sampling rate if PLL = 270.64MHz

Sample Rate (kHz)	I2SCDR	I2SDIV.DV	Sample Rate Error (%)
48	1	11	0.11
44.1	1	12	-0.11
32	0	33	0.11
24	1	22	0.11
22.05	1	24	-0.11
16	1	33	0.11
12	1	44	0.11
11.025	1	48	-0.11
8	1	66	0.11

For an EXCLK clock frequency, try to generate PLL frequencies as close to the frequencies listed in Table 12-9 as possible. Table 12-11 lists the PLL parameters and audio sample errors at different PLL frequencies for EXCLK at 12MHz.

Table 12-11 PLL parameters and audio sample errors for EXCLK=12MHz

PLL			Max Sample Rate Error
M	N	Freq (MHz)	
103	10	123.6	0.59%
49	4	147	0.31%
128	9	170.67	0.79%
157	8	235.5	0.87%
103	5	247.2	0.59%
65	3	260	0.82%
45	2	270	0.35%
203	9	270.67	0.12%
113	5	271.2	0.32%
187	8	280.5	0.75%
237	10	284.4	0.81%
49	2	294	0.31%
178	7	305.14	0.67%
53	2	318	0.60%
302	11	329.45	0.70%

256	9	341.33	0.79%
318	11	346.91	0.88%
206	7	353.14	0.90%
299	10	358.8	0.69%
247	8	370.5	0.55%
351	11	382.91	0.55%
230	7	394.29	0.27%

The BIT_CLK should be stopped temporary when change the divider settings, or when change BIT_CLK source (from internal or external), to prevent clock glitch. Register I2SCR.STPBK is provided to assist the task. When I2SCR.STPBK = 1, BIT_CLK is disabled no matter whether it is generated internally or inputted from the external source. The operation flow is described in following.

- 1 Stop all replay/record by clear AICCR.ERPL and AICCR.EREC.
- 2 Polling I2SSR.BSY till it is 0.
- 3 Stop the BIT_CLK by write 1 to I2SCR.STPBK.
- 4 Operations concerning BIT_CLK.
- 5 Resume the BIT_CLK by write 0 to I2SCR.STPBK.

12.4.11 Interrupts

The following status bits, if enabled, interrupt the processor:

- Receive FIFO Service (AICSR.RFS). It's also DMA Request.
- Transmit FIFO Service (AICSR.TFS). It's also DMA Request.
- Transmit Under-Run (AICSR.TUR).
- Receive Over-Run (AICSR.ROR).
- Command Address and Data Transmitted, AC-link only (ACSR.CADT).
- External CODEC Registers Status Address and Data Received, AC-link only (ACSR.SADR).
- External CODEC Registers Read Status Time Out, AC-link only (ACSR.RSTO).

For further details, see the corresponding register description sections.

12.5 SPDIF Guide

12.5.1 Set SPDIF clock frequency

Set SPDIF clock frequency is as same as i2s clock.

12.5.2 PCM audio mode operation (Reference IEC60958)

- 1 Set SPCFG1 and SPCFG2 to configure SPDIF transmitter.
 - a Set SPCFG2.CON_PRO to 0 to choose consumer mode.
 - b Set SPCFG2.AUDIO_N to 0 to choose linear PCM audio data mode.

- c Set SPCFG1.XXX to configure SPDIF.
 - d Set SPCFG2.XXX to configure SPDIF.
- 2 Set SPCTRL.DMA_EN to choose DMA mode or CPU mode.
- 3 Set SPCTRL.SIGN_N to choose whether to transfer the most significant bit by toggle or not.
- 4 Set SPCTRL.SFT_RST to 1 reset FIFO.
- 5 Wait SPCTRL.SFT_RST set to be set 0 by hardware.
- 6 Set SPCTRL.M_TRIG and SPCTRL.M_FFUR to enable or disable the interrupt.
- 7 Set SPCTRL.INVALID 1 or 0 to set the V bit of sub-frame.
- 8 Set SPENA.SPEN to 1 to Enable SPDIF to transmitter.

12.5.3 Non-PCM mode operation (Reference IEC61937)

- 1 Set SPCFG1 and SPCFG2 to configure SPDIF transmitter.
 - a Set SPCFG2.CON_PRO to 0 to choose consumer mode.
 - b Set SPCFG2.AUDIO_N to 1 to choose non-PCM mode.
 - c Set SPCFG1.SRC_NUM to 0.
 - d Set SPCFG1.CH1_NUM to 0.
 - e Set SPCFG1.CH2_NUM to 0.
 - f Set SPCFG2.PRE to 0.
 - g Set SPCFG2.CH_MD to 0.
 - h Set SPCFG2.ORG_FRQ to 0.
 - i Set SPCFG2.SAMPL_WL to 0.
 - j Set SPCFG2.MAX_WL to 0.
 - k Set SPCFG1.XXX to configure SPDIF.
 - l Set SPCFG2.XXX to configure SPDIF.
- 2 Set SPCTRL.DMA_EN to choose DMA mode or CPU mode.
- 3 Set SPCTRL.SIGN_N to choose whether to transfer the most significant bit by toggle or not.
- 4 Set SPCTRL.SFT_RST to 1 reset FIFO.
- 5 Wait SPCTRL.SFT_RST to be set to 0 by hardware.
- 6 Set SPCTRL.M_TRIG and SPCTRL.M_FFUR to enable or disable the interrupt.
- 7 Set SPCTRL.INVALID 1 or 0 to set the V bit of sub-frame.
- 8 Set SPENA.SPEN to 1 to Enable SPDIF to transmitter.

12.5.4 Disable operation

- 1 Set SPENA.SPEN to 0 to disable SPDIF to transmitter.
- 2 Wait SPSTATE.BUSY to be set to 0 by hardware.
- 3 You can do other operation.

13 PCM Interface

13.1 Overview

The PCM has the following features:

- Data starts with the frame PCMSYN or one PCMCLK later
- Support three modes of operation for PCM
 - Short frame sync mode
 - Long frame sync mode
 - Multi-slot mode
- Data is transferred and received with the MSB first
- Support master mode and slave mode
- The PCM serial output data, PCMDOUT, is clocked out using the rising edge of the PCMSCLK
- The PCM serial input data, PCMDIN, is clocked in on the falling edge of the PCMSCLK
- 8/16 bit sample data sizes supported
- DMA transfer mode supported
- Two FIFOs for transmit and receive respectively with 16 samples capacity in every direction

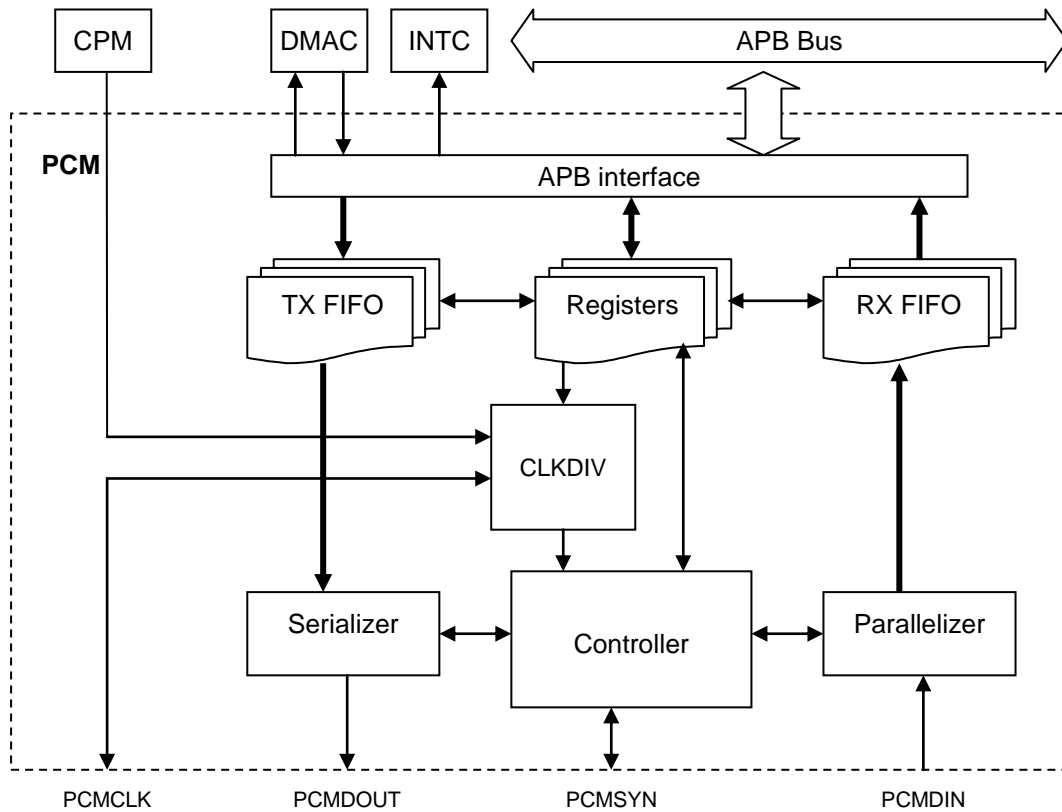
13.2 Pin Description

There are all 4 pins used to connect between PCM interface and an external device. They are listed and described in following table Table 13-1.

Table 13-1 PCM Interface Pins Description

Name	I/O	Description
PCMCLK	Input/Output	PCM Serial clock Line signal input/output
PCMSYN	Input/Output	PCM sync signal input/output
PCMDOUT	Output	PCM Serial data output
PCMDIN	Input	PCM Serial data input

13.3 Block Diagram



13.4 Register Description

Table 13-2 PCM Registers Description

Name	Description	RW	Reset Value	Address	Size
PCMCTL0	PCM Control Register	RW	0x00000000	0x10071000	32
PCMCFG0	PCM Configure Register	RW	0x00000110	0x10071004	32
PCMDP0	PCM FIFO Data Port Register	RW	0x00000000	0x10071008	32
PCMINTC0	PCM Interrupt Control Register	RW	0x00000000	0x1007100c	32
PCMINTS0	PCM Interrupt Status Register	RW	0x00000100	0x10071010	32
PCMDIV0	PCM Clock Divide Register	RW	0x00000001	0x10071014	32

13.4.1 PCM Control Register (PCMCTL)

PCMCTL0																0x10071000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																ERDMA	ETDMA	LSMP	ERPL	EREC	FLUSH	RST	SYNEN	CLKEN	PCMEN						
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW						
31:10	Reserved	Writing has no effect, read as zero.	R						
9	ERDMA	Receive DMA Enable. This bit is used to enable or disable the DMA during receiving audio data. <table border="1" style="margin-left: 20px;"> <tr> <th>ERDMA</th> <th>Receive DMA</th> </tr> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </table>	ERDMA	Receive DMA	0	Disabled.	1	Enabled.	RW
ERDMA	Receive DMA								
0	Disabled.								
1	Enabled.								
8	ETDMA	Transmit DMA Enable. This bit is used to enable or disable the DMA during transmit audio data. <table border="1" style="margin-left: 20px;"> <tr> <th>ETDMA</th> <th>Transmit DMA</th> </tr> <tr> <td>0</td> <td>Disabled.</td> </tr> <tr> <td>1</td> <td>Enabled.</td> </tr> </table>	ETDMA	Transmit DMA	0	Disabled.	1	Enabled.	RW
ETDMA	Transmit DMA								
0	Disabled.								
1	Enabled.								
7	LSMP	Select between play last sample or play ZERO sample in TX FIFO underflow. ZERO sample means sample value is zero. <table border="1" style="margin-left: 20px;"> <tr> <th>LSMP</th> <th>CODEC used</th> </tr> <tr> <td>0</td> <td>Play ZERO sample when TX FIFO underflow.</td> </tr> <tr> <td>1</td> <td>Play last sample when TX FIFO underflow.</td> </tr> </table>	LSMP	CODEC used	0	Play ZERO sample when TX FIFO underflow.	1	Play last sample when TX FIFO underflow.	RW
LSMP	CODEC used								
0	Play ZERO sample when TX FIFO underflow.								
1	Play last sample when TX FIFO underflow.								
6	ERPL	Enable Playing Back function. This bit is used to disable or enable the audio sample data transmitting. <table border="1" style="margin-left: 20px;"> <tr> <th>ERPL</th> <th>Description</th> </tr> <tr> <td>0</td> <td>PCM Playing Back Function is Disabled.</td> </tr> <tr> <td>1</td> <td>PCM Playing Back Function is Enabled.</td> </tr> </table>	ERPL	Description	0	PCM Playing Back Function is Disabled.	1	PCM Playing Back Function is Enabled.	RW
ERPL	Description								
0	PCM Playing Back Function is Disabled.								
1	PCM Playing Back Function is Enabled.								
5	EREC	Enable Recording Function. This bit is used to disable or enable the audio sample data receiving. <table border="1" style="margin-left: 20px;"> <tr> <th>EREC</th> <th>Description</th> </tr> <tr> <td>0</td> <td>PCM Recording Function is Disabled.</td> </tr> <tr> <td>1</td> <td>PCM Recording Function is Enabled.</td> </tr> </table>	EREC	Description	0	PCM Recording Function is Disabled.	1	PCM Recording Function is Enabled.	RW
EREC	Description								
0	PCM Recording Function is Disabled.								
1	PCM Recording Function is Enabled.								
4	FLUSH	FIFO Flush. Write 1 to this bit flush transmit/receive FIFOs to empty. Writing 0 to this bit has no effect and this bit is always reading 0.	W						
3	RST	Reset PCM. Write 1 to this bit reset PCM registers and FIFOs. Writing 0 to this bit has no effect and this bit is always reading 0.	W						
2	Reserved	Writing has no effect, read as zero.	R						
1	CLKEN	Enable the serial clock division logic. Must be HIGH for the PCM to operate.	RW						

0	PCMEN	Enable PCM function. This bit is used to enable or disable the PCM function.	RW		
				PCMENB	Description
				0	Disable PCM Controller.
1	Enable PCM Controller.				

13.4.2 PCM Configuration Register (PCMCFG)

PCMCFG0															0x10071004																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															SLOT	ISS	OSS	IMSBPOS	OMSBPOS	RFTH			TFTH			PCMMOD					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

Bits	Name	Description	RW										
31:15	Reserved	Writing has no effect, read as zero.	R										
14:13	SLOT	Controls the amount of valid PCM timeslot in one PCMSYN frame. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>SLOT</th> <th>Number of slot in one frame</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>1 slot</td> </tr> <tr> <td>01</td> <td>2 slot</td> </tr> <tr> <td>10</td> <td>3 slot</td> </tr> <tr> <td>11</td> <td>4 slot</td> </tr> </tbody> </table>	SLOT	Number of slot in one frame	00	1 slot	01	2 slot	10	3 slot	11	4 slot	RW
SLOT	Number of slot in one frame												
00	1 slot												
01	2 slot												
10	3 slot												
11	4 slot												
12	ISS	Input Sample Size. These bits reflect input sample data size to memory or register. The data sizes supported are: 8/16bits. The sample data is LSB-justified in memory/register. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>ISS</th> <th>Sample Size</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>8 bit</td> </tr> <tr> <td>1</td> <td>16 bit</td> </tr> </tbody> </table>	ISS	Sample Size	0	8 bit	1	16 bit	RW				
ISS	Sample Size												
0	8 bit												
1	16 bit												
11	OSS	Output Sample Size. These bits reflect output sample data size from memory or register. The data sizes supported are: 8/16 bits. The sample data is LSB-justified in memory/register. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>OSS</th> <th>Sample Size</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>8 bit</td> </tr> <tr> <td>1</td> <td>16 bit</td> </tr> </tbody> </table>	OSS	Sample Size	0	8 bit	1	16 bit	RW				
OSS	Sample Size												
0	8 bit												
1	16 bit												
10	IMSBPOS	Controls the position of the MSB bit in the serial input stream relative to the PCMSYN signal. 0: MSB is captured on the falling edge of PCMCLK during the same cycle that PCMSYNC is high 1: MSB is captured on the falling edge of PCMCLK during the cycle	RW										

		after the PCMSYNC is high	
9	OMSBPOS	Controls the position of the MSB bit in the serial output stream relative to the PCMSYN signal. 0: MSB sent during the same clock that PCMSYN is high 1: MSB sent on the next PCMSCLK cycle after PCMSYNC is high	RW
8:5	RFTH	Receive FIFO threshold for interrupt or DMA request. Determines when the RFS flags go active for the RXFIFO. When the sample number in receive FIFO, indicated by PCMINTS.RFL, is great than the threshold value, PCMINTS.RFS is set. Larger RFTH value provides lower DMA/interrupt request frequency but have more risk to involve receive FIFO overflow. The optimum value is system dependent.	RW
4:1	TFTH	Transmit FIFO threshold for interrupt or DMA request. Determines when the TFS flags go active for the TXFIFO. When the sample number in transmit FIFO, indicated by PCMINTS.TFL, is less than the threshold value, PCMINTS.TFS is set. Smaller TFTH value provides lower DMA/interrupt request frequency but have more risk to involve transmit FIFO underflow. The optimum value is system dependent.	RW
0	PCMMOD	PCM mode select. 0: Master mode; 1: Slave mode.	RW

13.4.3 PCM FIFO DATA PORT REGISTER (PCMDP)

	PCMDP0		0x10071008																														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	DATA																																
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	DATA	FIFO port. When write to it, data is push to the transmit FIFO. When read from it, data is pop from the receiving FIFO.	RW

13.4.4 PCM INTERRUPT CONTROL REGISTER (PCMINTC)

	PCMINTCR0		0x1007100c																													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

14	RSTS	Soft reset / flush state. 0: Nothing / reset or flush operation has completed 1: reset or flush operation has not completed	R								
13:9	TFL	Transmit FIFO Level. The bits indicate the amount of valid PCM data in Transmit FIFO.	R								
8	TFS	Transmit FIFO Service Request. This bit indicates that transmit FIFO level exceeds TFL threshold which is controlled by PCMCFG.TFTH When TFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting. <table border="1" data-bbox="582 607 1281 734"> <thead> <tr> <th>TFS</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Transmit FIFO level exceeds TFL threshold.</td> </tr> <tr> <td>1</td> <td>Transmit FIFO level at or below TFL threshold.</td> </tr> </tbody> </table>	TFS	Description	0	Transmit FIFO level exceeds TFL threshold.	1	Transmit FIFO level at or below TFL threshold.	RW		
TFS	Description										
0	Transmit FIFO level exceeds TFL threshold.										
1	Transmit FIFO level at or below TFL threshold.										
7	TUR	Transmit FIFO Under Run. This bit indicates that transmit FIFO has or has not experienced an under-run. <table border="1" data-bbox="582 822 1281 1117"> <thead> <tr> <th>TUR</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td rowspan="2">0</td> <td>When read, indicates under-run has not been found.</td> </tr> <tr> <td>When write, clear itself.</td> </tr> <tr> <td rowspan="2">1</td> <td>When read, indicates data has even been read from empty transmit FIFO.</td> </tr> <tr> <td>When write, not effects.</td> </tr> </tbody> </table>	TUR	Description	0	When read, indicates under-run has not been found.	When write, clear itself.	1	When read, indicates data has even been read from empty transmit FIFO.	When write, not effects.	RW
TUR	Description										
0	When read, indicates under-run has not been found.										
	When write, clear itself.										
1	When read, indicates data has even been read from empty transmit FIFO.										
	When write, not effects.										
6:2	RFL	Receive FIFO Level. The bits indicate the amount of valid PCM data in Receive FIFO.	R								
1	RFS	Receive FIFO Service Request. This bit indicates that receive FIFO level is or not below RFL threshold which is controlled by PCMCFG.RFTH. When RFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting. <table border="1" data-bbox="582 1370 1281 1498"> <thead> <tr> <th>RFS</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Receive FIFO level below RFL threshold.</td> </tr> <tr> <td>1</td> <td>Receive FIFO level at or above RFL threshold.</td> </tr> </tbody> </table>	RFS	Description	0	Receive FIFO level below RFL threshold.	1	Receive FIFO level at or above RFL threshold.	RW		
RFS	Description										
0	Receive FIFO level below RFL threshold.										
1	Receive FIFO level at or above RFL threshold.										
0	ROR	Receive FIFO Over Run. This bit indicates that receive FIFO has or has not experienced an overrun. <table border="1" data-bbox="533 1585 1281 1839"> <thead> <tr> <th>ROR</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td rowspan="2">0</td> <td>When read, indicates over-run has not been found.</td> </tr> <tr> <td>When write, clear itself.</td> </tr> <tr> <td rowspan="2">1</td> <td>When read, indicates data has even been written to full receive FIFO.</td> </tr> <tr> <td>When write, not effects.</td> </tr> </tbody> </table>	ROR	Description	0	When read, indicates over-run has not been found.	When write, clear itself.	1	When read, indicates data has even been written to full receive FIFO.	When write, not effects.	RW
ROR	Description										
0	When read, indicates over-run has not been found.										
	When write, clear itself.										
1	When read, indicates data has even been written to full receive FIFO.										
	When write, not effects.										

13.4.6 PCM CLOCK DIVIDE REGISTER (PCMDIV)

PCMDIV0																0x10071014																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																SYNL				SYNDIV				CLKDIV							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:27	Reserved	Writing has no effect, read as zero.	R
16:11	SYNL	Controls the length that the PCMSYN based upon the PCMCLK. The length of PCMSYN = (SYNL + 1) * PCMCLK cycle.	RW
10:6	SYNDIV	Controls the frequency of the PCMSYN signal based upon the PCMCLK. $PCMSYN = PCMCLK / 8 (SYNDIV + 1)$.	RW
5:0	CLKDIV	PCMCLK clock divider value minus 1. Controls the divider used to create the PCMCLK based upon the CPM_PCM_SYSCCLK. $PCMCLK = CPM_PCM_SYSCCLK / (CLKDIV + 1)$.	RW

13.5 PCM Interface Timing

The following figures show the timing relationship for the PCM transfers, Note in all cases. In master mode, the PCMCLK is derived from dividing the input clock, CPM_PCM_SYSCCLK, and the PCMSYN is divided depended on the PCMCLK. In slave mode, the PCMCLK and PCMSYN are input from the external device. Data is sampled on the falling edge of the PCMCLK and sent out on the rising edge of the PCMCLK. The PCMSYN signal determines when the next data sample is to be transferred between the controller and the external device. Also, the PCMSYN signal as seen in the figure can be one bit time or a long bit time controlled by PCMDIV.SYNL. The PCMSYN frequency controlled by PCMDIV.SYNDIV is usually the sample rate. There are some variations controlled by PCMCFG.ISS, PCMCFG.OSS and PCMCFG.SLOT to accommodate 8 / 16bit sample sizes and multi-slot transmission.

13.5.1 Short Frame SYN Timing

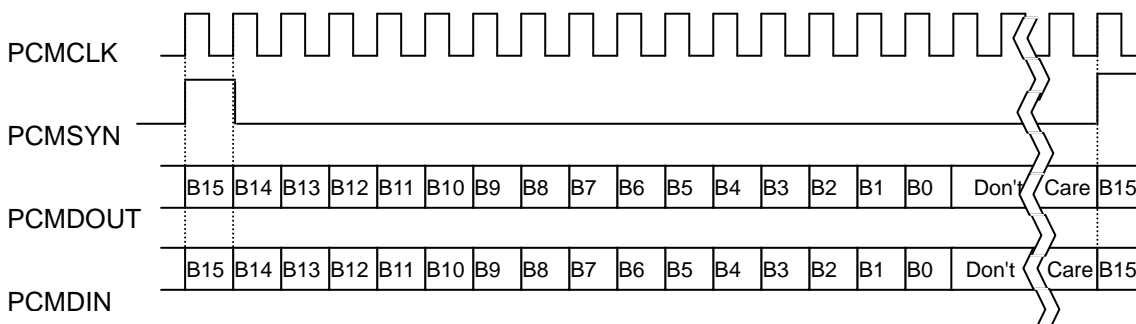


Figure 13-1 Short Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 13-1 shows a PCM transfer with the MSB configured to be coincident with the

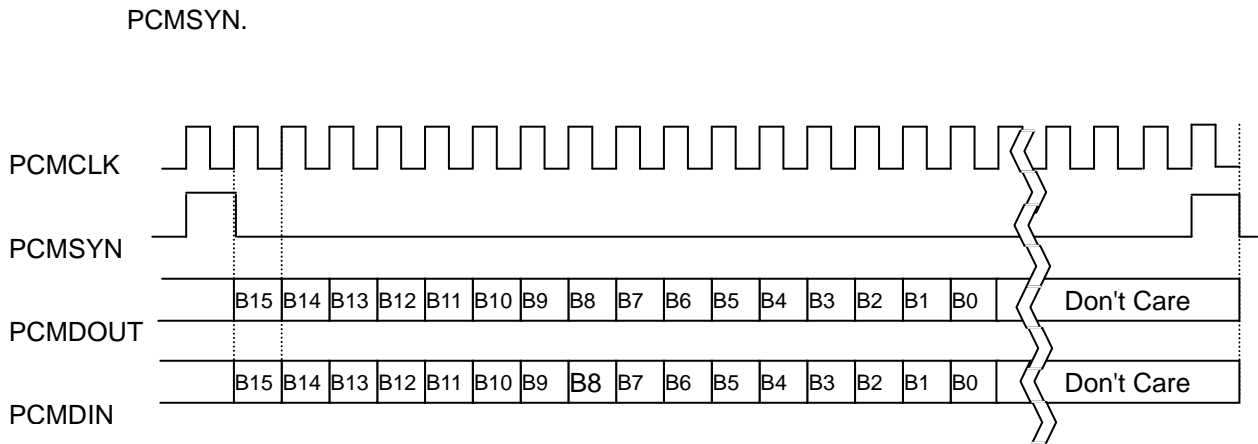


Figure 13-2 Short Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 13-2 shows a PCM transfer with the MSB configured one shift clock after the PCMSYN.

13.5.2 Long Frame SYN

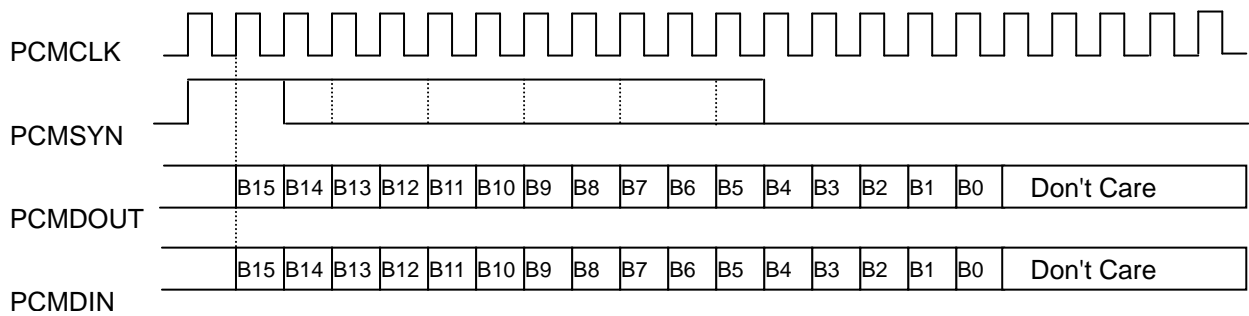


Figure 13-3 Long Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 13-3 shows a PCM transfer with the MSB configured one shift clock after the PCMSYN.

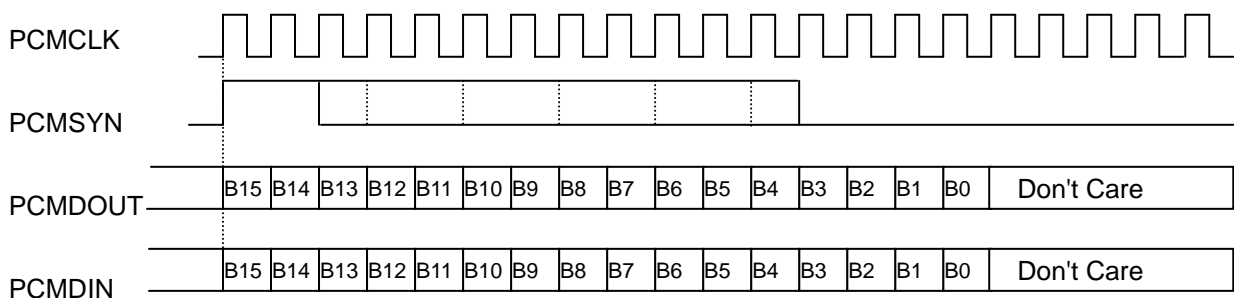


Figure 13-4 Long Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 13-4 shows a PCM transfer with the MSB configured to be coincident with the PCMSYN.

13.5.3 Multi-Slot Operation

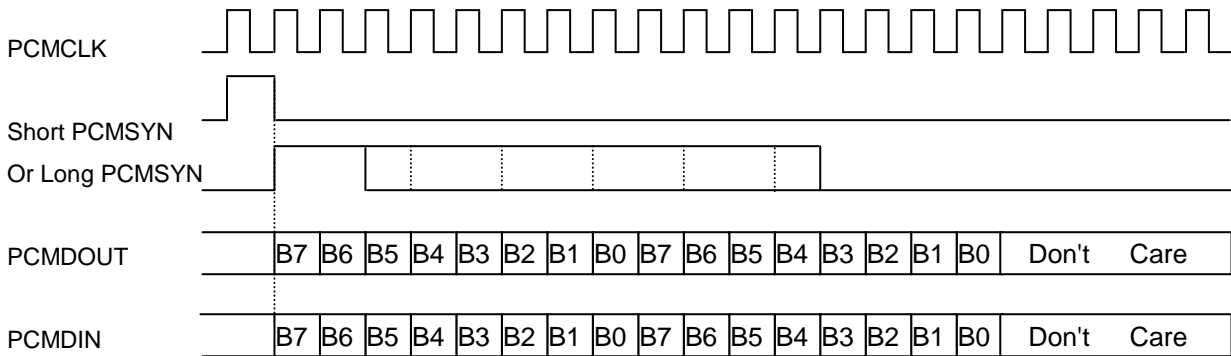
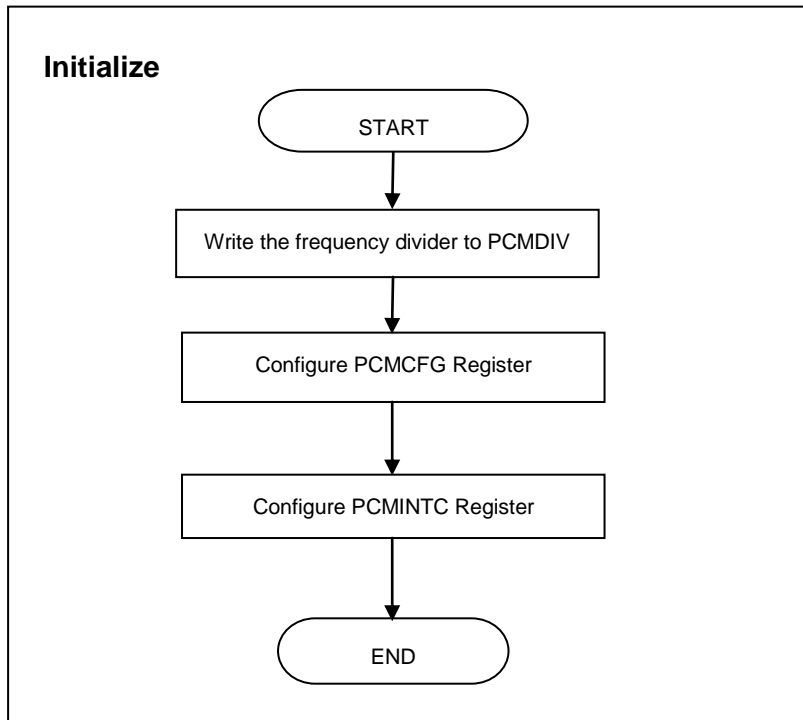


Figure 13-5 Multi-Slot Frame SYN Timing (Shown with two Slots and 8bit Sample)

13.6 PCM Operation

13.6.1 PCM Initialization

At power-on or other hardware reset (WDT and etc), PCM is disabled. Software must initiate PCM after power-on or reset.



For further details, see the corresponding register description sections.

13.6.2 Audio Replay

Outgoing audio sample data is written to PCM transmit FIFO from processor via store instruction or from memory via DMA. PCM then takes the data from the FIFO, serializes it, and sends it over the serial wire PCMDOUT to an external DEVICE.

The audio transmission is enabled automatically when the PCM is enabled by set PCMCTL.PCMEN. And PCMCTL.ERPL must be 1. If PCMCTL.ERPL is 0, value of zero is sent to external DEVICE even if there are samples in transmit FIFO. At least one audio sample data in the transmit FIFO. If the transmit FIFO is empty, value of zero or last sample depends on AICFR.LSMP, is send to external DEVICE even if PCMCTL.ERPL is 1.

Here is the audio replay flow:

- 1 Configure the external DEVICE as needed.
- 2 Initialize PCM and configure the register.
- 3 Write 1 to PCMCTL.PCMEN and PCMCTL.CLKEN.
- 4 Fill sample data to the transmit FIFO. Repeat this till finish all sample data. In this procedure, please control the FIFO to make sure no FIFO under-run and other errors happen. When the transmit FIFO under-run, noise or pause may be heard in the audio replay, PCMINTS.TUR is 1, and if PCMINTC.ETUR is 1, PCM issues an interrupt. Please reference to 13.6.4 for detail description on FIFO.
- 5 Write 1 to PCMCTL.ERPL. It is suggested that at least a frame of PCM data is pre-filled in the transmit FIFO to prevent FIFO under-run flag (PCMINTS.TUR).
- 6 Waiting for PCMINTS.TFL change to 0. So that all samples in the transmit FIFO has been replayed, then we can have a clean start and write 0 to PCMCTL.ERPL.

13.6.3 Audio Record

Incoming audio sample data is received from PCMDIN serially and converted to parallel word and stored in PCM receive FIFO. Then the data can be taken from the FIFO to processor via load instruction or to memory via DMA.

The audio recording is enabled automatically when the PCM is enabled by set PCMCTL.PCMEN, And PCMCTL.EREC must be 1. If PCMCTL.EREC is 0, the received data is discarded even if there are rooms in the receive FIFO. At least one room left in the receive FIFO. If the receive FIFO is full, the received data is discarded even if PCMCTL.EREC is 1.

Here is the audio record flow:

- 1 Configure the external DEVICE as needed.
 - a Initialize PCM and configure the register.
 - b Write 1 to PCMCTL.PCMEN and PCMCTL.CLKEN.
- 2 Write 1 to PCMCTL.EREC. Make sure there are rooms available in the receive FIFO before

set PCMCTL.EREC. Usually, it should empty the receive FIFO by fetch data from it before set PCMCTL.EREC.

- 3 Take sample data form the receive FIFO. Repeat this till the audio finished. In this procedure, please control the FIFO to make sure no FIFO over-run and other errors happen. When the receive FIFO over-run, same recorded audio samples will be lost, PCMINTS.ROR is 1, and if PCMINTC.EROR is 1, PCM issues an interrupt. Please reference to _13.6.4 for detail description on FIFO.
- 4 Write 0 to AICCR.EREC.
- 5 Take sample data from the receive FIFO until PCMINTS.RFL change to 0. So that all samples in the receive FIFO has been taken away, then we can have a clean start up next time. When the receive FIFO is empty, read from it returns zero.

13.6.4 FIFOs operation

PCM has two FIFOs, one for transmitting and one for receiving. The FIFOs are in 16 bits width and 16 entries depth, one entry for keep one sample regardless of the sample size. PCMDP.DATA provides the access point for processor/DMA to write to transmit FIFO and read from receive FIFO. One time access to PCMDP.DATA process one sample. The sample data should be put in LSB (Least Significant Bit) in memory or processor registers. For transmitting, bits exceed sample are discarded. For receiving, these bits are set to 0. Following figure illustrates the FIFOs access.

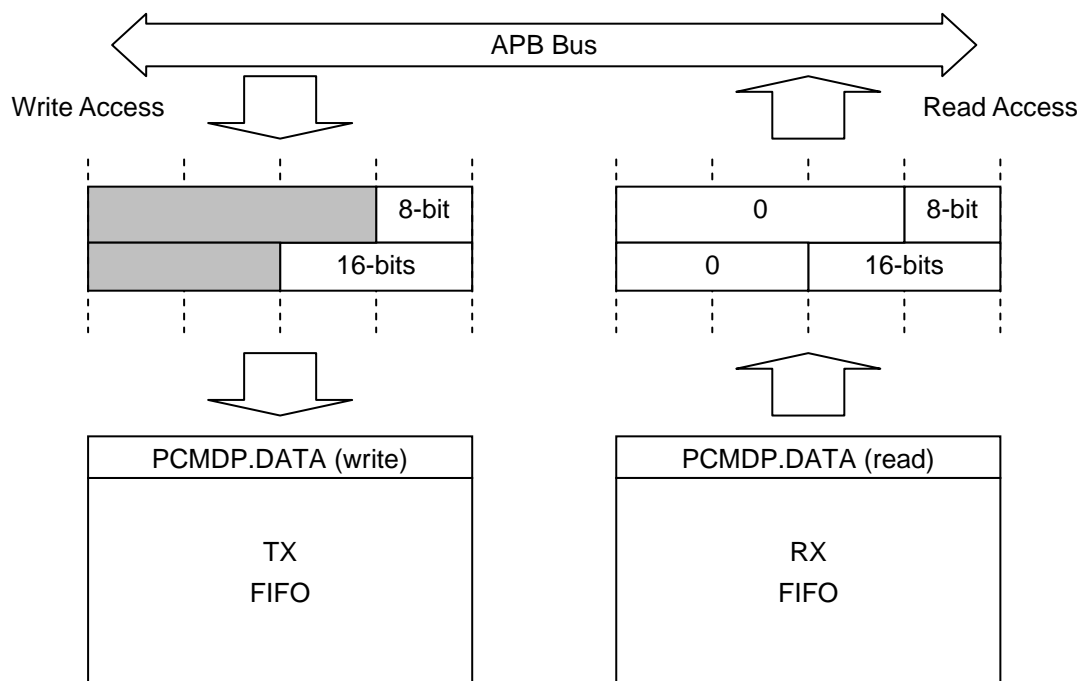


Figure 13-6 Transmitting/Receiving FIFO access via APB Bus

The software and bus initiator must guarantee the right sample placement at the bus.

In case of DMA bus initiator, One 16 bits sample occupies one 16-bits half word in memory, so 16-bits width DMA must be used. One 8-bits sample occupies one byte in memory, and use 8-bits width DMA.

13.6.5 Data Flow Control

There are three approaches provided to control/synchronize the incoming/outgoing data flow.

13.6.5.1 Polling and Processor Access

PCMINTS.RFL and PCMINTS.TFL reflect how many samples exist in receiving and transmitting FIFOs. Through read these register fields, processor can detect when there are samples in receiving FIFO and then load them from the RxFIFO, and when there are rooms in transmitting FIFO and then store samples to the TxFIFO.

Polling approach is in very low efficiency and is not recommended.

13.6.5.2 Interrupt and Processor Access

Set proper values to PCMCFG.TFTH and PCMCFG.RFTH, the FIFO interrupts trig thresholds. Set PCMINTC.ETFS and/or PCMINTC.ERFS to 1 to enable transmitting and/or receiving FIFO level trigger interrupts. When the interrupt found, it means there are rooms or samples in the TX or RX FIFO, and processor can store or load samples to or from the FIFO.

Interrupt approach is more efficient than polling approach.

13.6.5.3 DMA Access

To enable DMA operation, set PCMCTL.ERDMA and PCMCTL.ETDMA to 1 for transmit and receive respectively. It also needs to allocate two channels in DMA controller for data transmitting and receiving respectively. Please reference to DMAC spec for the details.

The PCMCFG.TFTH and PCMCFG.RFTH are used to set the transmitting and receiving FIFO level thresholds, which determine the issuing of DMA request to DMA controller. To respond the request, DMAC initiator and controls the data movement between memory and TX/RX FIFO.

13.6.6 PCM Serial Clocks and Sampling Frequencies

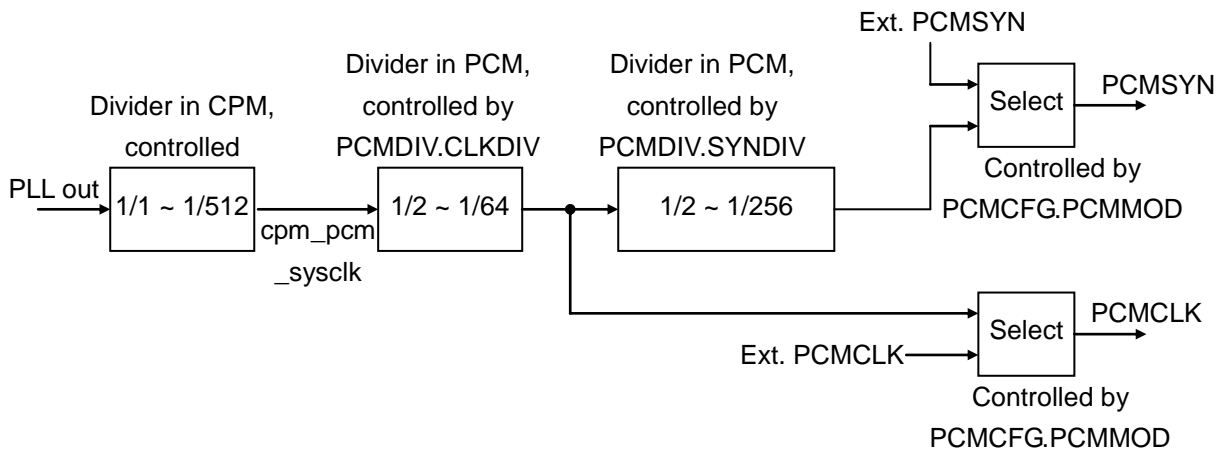


Figure 13-7 PCMCLK and PCMSYN generation scheme

13.6.7 Interrupts

The following status bits, if enabled, interrupt the processor:

- Receive FIFO Service (PCMINTS.RFS). It's also DMA Request.
- Transmit FIFO Service (PCMINTS.TFS). It's also DMA Request.
- Transmit Under-Run (PCMINTS.TUR).
- Receive Over-Run (PCMINTS.ROR).

For further details, see the corresponding register description sections.

14 Internal CODEC Interface

14.1 Overview

This chapter describes the embedded audio CODEC in the processor and related software **interfaces**.

This embedded CODEC is an I2S audio CODEC. AIC module is an interface to this CODEC for audio data replaying and recording. Several memory mapped registers **in AIC** are used to access this embedded CODEC, **and the CODEC's internal control and configure registers can be accessed by write/read these memory mapped registers using 12 MHz clock.**

14.2 Features

The following are internal CODEC features:

- 24 bits ADC and DAC
- Headphone load up to 16 Ohm
- Sample frequency supported: 8k, 11.025k, 12k, 16k, 22.05k, 24k, 32k, 44.1k, 48k, and 96k
- Stereo line input
- DAC to HP path: Power consumption: 17.6mW, THD: -65dB @17.6mW /16Ohm
- DAC to stereo line output path @10kOhm: SNR: 100dB A-Weighted, THD: -80dB @FS-1dB
- Line input to ADC path: SNR: 90dB A-Weighted, THD: -80dB @FS-1dB
- Separate power-down modes for ADC and DAC path with several shutdown modes
- Reduction of audible glitches systems: Pop Reduction system, Soft Mute mode
- Output short circuit protection
- Embedded low noise Linear Regulator
- 3 MIC in path or 3lin in path Maximum (total input share to 2 path at same time)

TBD = parameter or document section to be defined later on

TBC = parameter or document section subject to change

TO BE COMPLETED = section to be filled or subject to change

14.2.1 Signal Descriptions

CODEC has **maximum** 13 analog signal IO pins and 4 power pins on chip. They are listed and described in the following table.

Table 14-1 CODEC signal IO pin description

Pin Names	IO	Pin Description	Power
AIP1, AIN1	AI	Single-ended or differential analog input.	AVDCDC
AIP2	AI	Single-ended analog input.	AVDCDC
AIP3	AI	Single-ended analog input.	AVDCDC
MICBIAS1	AO	Microphone bias.	AVDCDC
AOLOP	AO	Differential line output, positive pin.	AVDCDC
AOLON	AO	Differential line output, negative pin.	AVDCDC
AOHPL	AO	Left headphone single-ended analog output.	AVDCDC
AOHPR	AO	Right headphone single-ended analog output.	AVDCDC
VCAP	AO	Voltage Reference Output. An 10 μ F ceramic or tantalum capacitor in parallel with a 0.1 μ F ceramic capacitor attached from this pin to AVSCDC eliminates the effects of high frequency noise.	AVDCDC
VREFP	AO	Internal nLR output	AVDCDC
AVSHP	P	Headphone amplifier ground.	-
AVDCDC	P	CODEC analog power, 3.3V, internal nLR input.	-
AVSCDC	P	CODEC analog ground, inter signal VREFN.	-
HPSENSE	AI	Headphone jack sense.	AVDCDC
DMIC_IN	DI	Digital microphone data input pin.	AVDCDC
DMIC_CLK	DO	Digital microphone clock output pin.	AVDCDC

NOTES:

- 1 AVDCDC= 3.3v (typ).
- 2 Internal signal VREFN is connected to AVSCDC, Internal power supply AVS is connected to AVSCDC.
- 3 Please refer to data sheet of the chip for details.
- 4 DMIC_IN is 'GPIO : PB18' , MIC_CLK is 'GPIO : PB19'. Please refer to GPIO specification for these pins operating.

14.2.2 Block Diagram

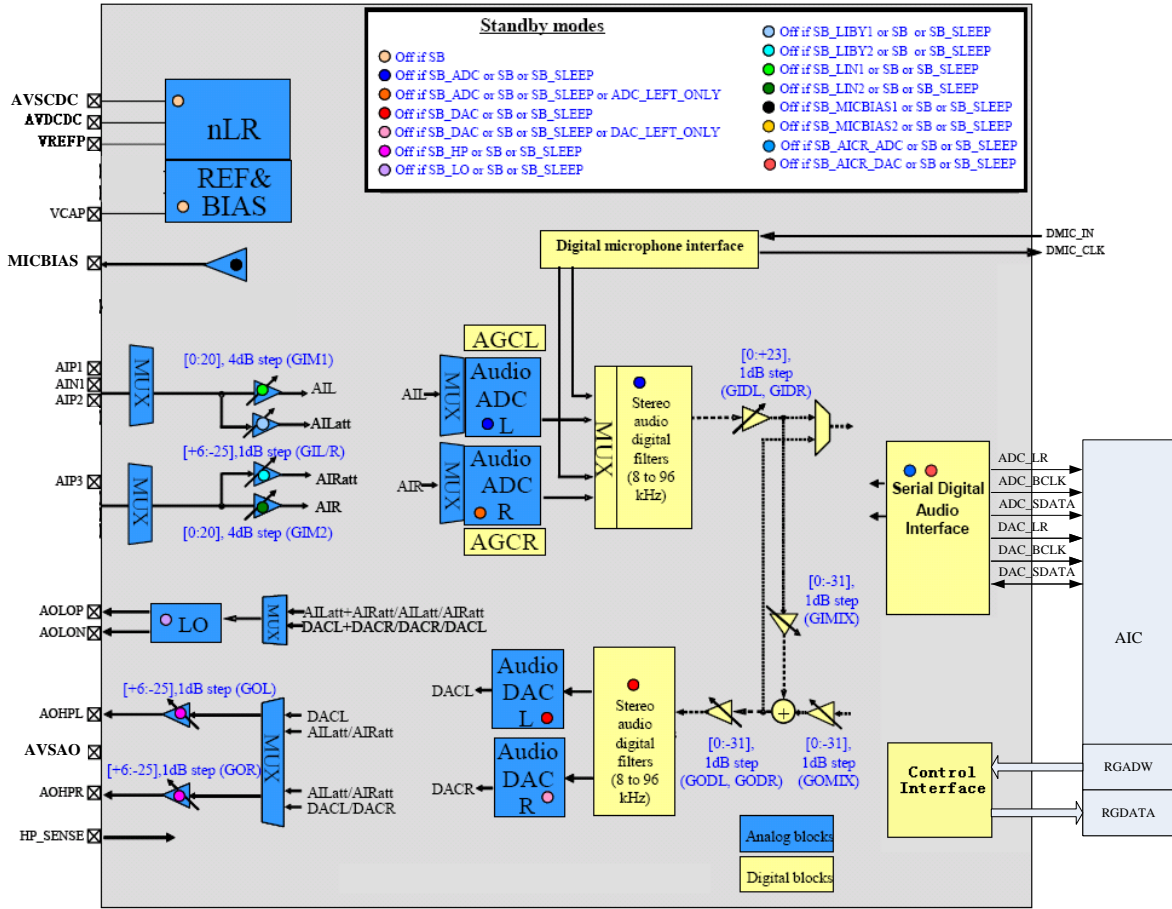


Figure 14-1 CODEC block diagram

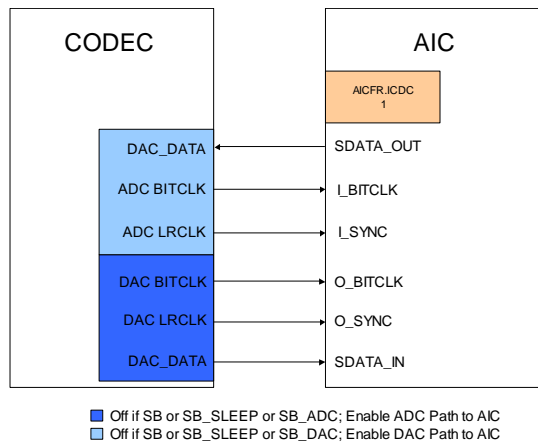
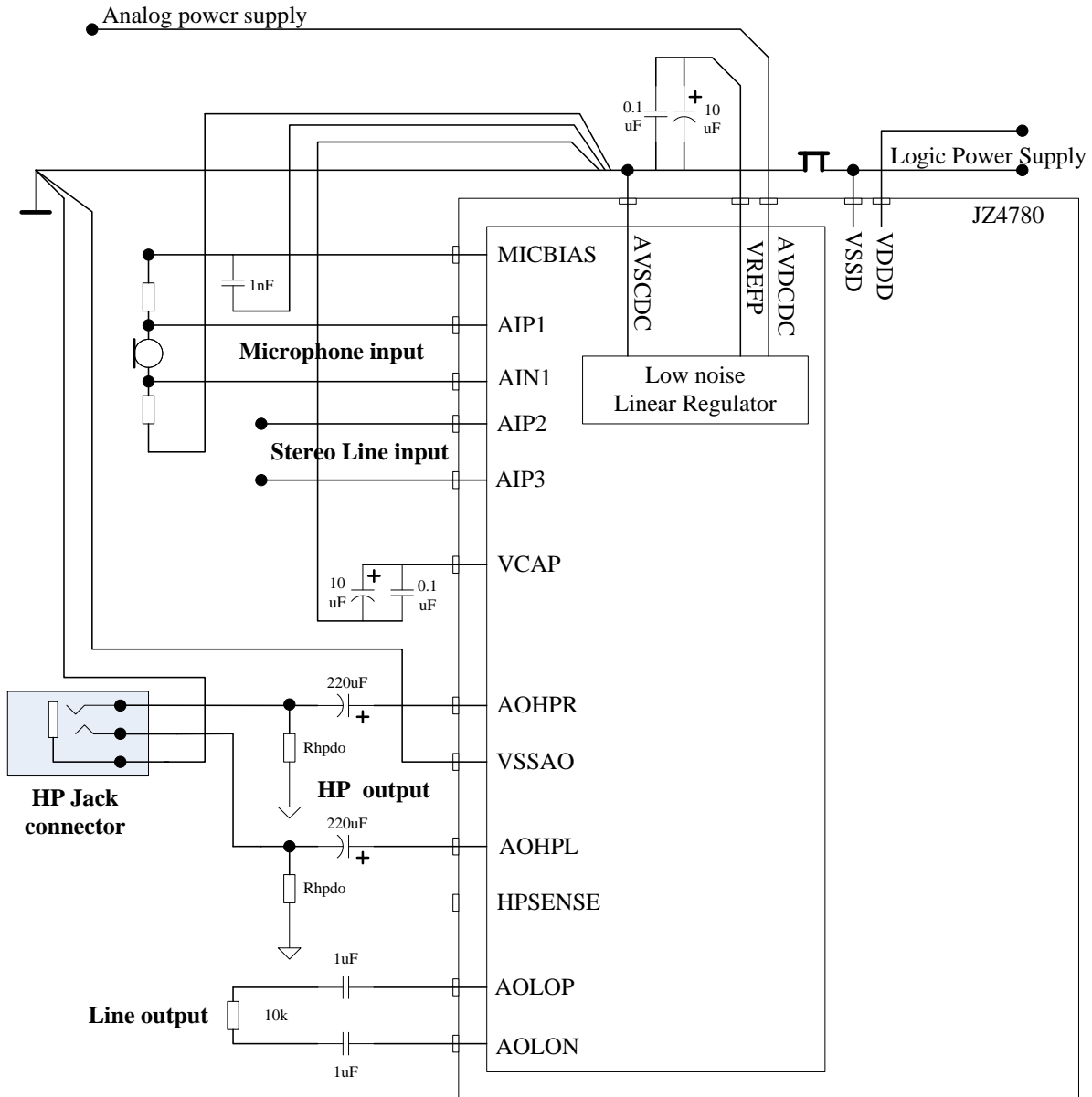


Figure 14-2 Internal CODEC works with AIC

14.2.3 Application schematic



- Note:
1. The Rhpdo value is 470 Ohm, it use to prevent pop-up noise.
 2. The single-ended/differential input port AIP1/AIN1 and single-ended input port AIP2/AIP3 can be configure to microphone input or line input by software.
 3. VREFP/VCAP/VREFP each of them requires connecting decoupling capacitors (0.1uF) between the pads VREFP/VCAP/VREFP and AVSCDC. This ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch)

14.3 Mapped Register Descriptions

The internal CODEC software interface includes 2 registers. They are mapped to IO memory address space of AIC module so that program can access them to control the operations of the CODEC.

Table 14-2 Internal CODEC Mapped Registers Description (AIC Registers)

Name	Description	RW	Reset value	Address	Size
RGADW	Address, data in and write command for accessing to internal registers of internal embedded CODEC.	RW	0x00000000	0x100200A4	32
RGDATA	The data read out and interrupt request status of Internal registers data in the internal embedded CODEC.	R	0x00000000	0x100200A8	32

NOTES:

- 1 All these registers are AIC Registers, because they are mapped to AIC IO memory address.
- 2 RGADW contains data, address and write command to the internal registers of the internal CODEC.
- 3 RGDATA returns the internal register value of the internal CODEC and interrupt request status.

14.3.1 CODEC internal register access control (RGADW)

RGADW contains address, data and write command to the internal registers of the internal embedded CODEC.

RGADW																0x100200A4																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	ICRST	Reserved														RGWR	Reserved	RGADDR								RGDIN							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW
31	ICRST	If set '1' to this bit internal codec will reset and then clear to '0' to end reset.	RW
30:17	Reserved	Writing has no effect, read as zero.	R
16	RGWR	Writing 1 to this bit means issuing a write command to CODEC's internal register. This bit keeps 1 by AIC before the current write process is finished. A register read or a new register write cannot be issued before the previous write process is finished. In another word, RGADW should not be written before RGADW.RGWR becomes 0.	RW

		A write process takes up to the sum of 0.17us and 1 PCLK cycle. Writing 0 to this bit is ignored.	
15	Reserved	Writing has no effect, read as zero.	R
14:8	RGADDR	When it issues a writing command to CODEC's internal register, i.e.RGWR=1, this field specifies the corresponding CODEC register's address. In addition, this field also decides the address of the CODEC register's data out (The data is mapped to RGDATA. RGDOOUT) at any time.	RW
7:0	RGDIN	When it issues a write command to CODEC's internal register, i.e. RGWR=1, this field contains the data to be written to the CODEC register.	RW

NOTES:

- 1 It is strongly suggested verifying the data (using read RGDATA below) after writing the data to internal register of CODEC. When verifying RGDATA. RGDOOUT right (RGDATA. RGDOOUT is the return data from internal CODEC register of the corresponding address), the writing process is finished.
- 2 Please notice that AIC needs SYS_CLK (please refer to AIC spec), when write new value to or read from CODEC internal registers.

14.3.2 CODEC internal register data output (RGDATA)

RGDATA returns the register value of the internal embedded CODEC and interrupt request status.

RGDATA																0x100200A8																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserve															IRQ	RGDOOUT															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW						
31:9	Reserved	Writing has no effect, read as zero.	R						
8	IRQ	This field returns the internal embedded CODEC's interrupt request. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>IRQ</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No CODEC's interrupt request found.</td> </tr> <tr> <td>1</td> <td>CODEC's interrupt request is pending.</td> </tr> </tbody> </table>	IRQ	Description	0	No CODEC's interrupt request found.	1	CODEC's interrupt request is pending.	R
IRQ	Description								
0	No CODEC's interrupt request found.								
1	CODEC's interrupt request is pending.								
7:0	RGDOOUT	This field returns the value of the register in internal embedded CODEC. (RGADW.RGADDR field specifies the CODEC register's address)	R						

NOTE: AIC needs SYS_CLK (please refer to AIC spec), when write new value to or read from CODEC internal registers.

14.4 Operation

The internal embedded CODEC is controlled **through** its internal registers. These registers **data** can be accessed through memory-mapped registers, RGADW and RGDATA. AIC's BITCLK and SYNC are **from** the CODEC and is controlled by CKCFG.SELAD register. The audio data transferring, i.e. audio replaying and recording, is done by AIC. AIC still **plays** the role of I2S controller. We will refer to many AIC operations and registers in the following audio operation descriptions, please reference to AIC spec for the details.

This is a guide for software.

14.4.1 Access to internal registers of the embedded CODEC

The embedded CODEC is controlled through its internal registers. RGADW and RGDATA are used to write to and read from these registers. Here are some examples.

Example 1. Write to a CODEC internal register.

Step 1: RGADW.RGWR == 0.

Step 2: If not, go to step 1.

Step 3: Write to RGADW and make it.

RGADW.RGDIN = <data to be written to the register>.

RGADW.RGADDR = <the register's address >.

Step 4: Write to RGADW to commit the writing operation.

RGADW.RGWR = 1.

Example 2. Read from a CODEC internal register.

Step 1: RGADW.RGWR == 0.

Step 2: If not, go to step 1.

Step 3: write to RGADW and make it.

RGADW.RGWR = 0.

RGADW.RGDIN = <don't care>.

RGADW.RGADDR = <the register's address>.

Step 4: read RGDATA.DOUT, which returns the register's content.

14.4.2 CODEC controlling and typical operations

This section is **about** some typical operations. We assume the power supply of CODEC is on, and CODEC is in STANDBY mode.

Before **performing** any of these operations, make sure AIC is configured properly as list below:

1 Make AIC to use internal CODEC mode:

AICFR.ICDC = 1; Use internal CODEC.

AICFR.AUSEL = 1; Use I2S mode.

AICFR.BCKD = 0; CODEC input BIT_CLK to AIC.

- AICFR.SYNCD = 0; CODEC input SYNC to AIC.
- I2SCR.AMSL = 1; Use I2S operation mode.
- I2SCR.ESCLK = 1; Open SYS_CLK to internal CODEC. (if using PLL Clock)
- 2 Make sure AICCR.FLUSH = 0; AICFR.RST = 0; AICCR.ENLBF = 0.
- 3 Clear AICSR.ROR, AICSR.TUR, AICSR.RFS, AICSR.TFS to 0.
- 4 Set proper value to AICCR.M2S; AICCR.ENDSW; AICCR.ASVTSU.
- 5 Set AICFR.ENB to 1; Open AIC.

When using DMA mode, configure AICFR.RFTH, AICCR.RDMS or AICFR.TFTH, AICCR.TDMS. Configure TX-FIFO and interrupt means setting proper value to AICFR.TFTH, clear AICCR.ETFS to 0, and clear AICCR.ETUR to 0.

Configure RX-FIFO and interrupt means setting proper value to AICFR.RFTH, clear AICCR.ERFS to 0 and clear AICCR.EROR to 0.

When configure interrupt, software must handle all the interrupts. So all interrupts are recommended disabled as shown above.

CODEC shares the interrupt with AIC module.

The register or register bit of CODEC will use the same form as the Mapped registers, but software should use the method in the section [“Mapped Register Descriptions”](#) to access this registers.

More details are listed in the CODEC guide.

14.4.3 Power saving

There are many power modes in CODEC. In every working mode, it should close **unused** stages (parts) of CODEC for saving power.

The power diagram is shown in “CODEC Power Diagram”; please refer to [“CODEC Operating modes”](#).

14.4.4 Pop noise and the reduction of it

[Please refer to “Ramping system note” and “Anti-pop operation sequences” for details.](#)

14.4.4.1 Reference open step

- 1 Init play.

Step 0: Open DMA and two AIC modules Clocks in CPM.CLKGR.

Step 1: Configure AIC as slave and using internal CODEC mode.

AICFR.ICDC = 1; Use internal CODEC.

AICFR.AUSEL = 1; Use I2S mode.

AICFR.BCKD = 0; CODEC input BIT_CLK to AIC.

AICFR.SYNCD = 0; CODEC input SYNC to AIC.

I2SCR.AMSL = 1; Use I2S operation mode.

I2SCR.ESCLK = 1; Open SYS_CLK to internal CODEC.

Step 2: Configure DMA as slave mode using internal CODEC.

2 Open.

Step 0: Enable DMA Channel Clock.

Step 1: Configure AIC sample size and sample rate. Configure AIC Output FIFO Threshold.

Step 2: Configure DMA.

Step 3: Configure CODEC.

3 Write.

Step 0: Enable DMA Channel Clock.

Step 1: Configure AIC.

Step 2: Configure DMA.

Step 3: Configure CODEC.

4 Read.

Step 0: Enable DMA Channel Clock.

Step 1: Configure AIC.

Step 2: Configure DMA.

Step 3: Configure CODEC.

5 Close.

6 End.

NOTES:

- 1 SB_DAC Control the internal OBIT_CLK from CODEC to AIC, First turn it on when write data (replay).
- 2 SB_ADC Control the internal IBIT_CLK from CODEC to AIC, First turns it on when read data (record).

14.5 Timing parameters

Parameter	Condition	Min.	Typ.	Max.	Unit
Tsbyu	Cext = 10uF/100nF +/-20%		250	600	ms
Tshd_adc	Cext = 10uF/100nF +/-20%		200		ms
Tshd_dac	Cext = 10uF/100nF +/-20%		400	900	ms
Tr, Tf (all inputs)	All modes			5	ns
Tr, Tf (all outputs)	All modes			5	ns

NOTES:

- 1 Tsbyu is the reference wake-up time after complete power down.
- 2 Tshd_adc is the ADC wake-up time after sleep mode.
- 3 Tshd_dac is DAC wake-up time after sleep mode.

14.6 C parameters

Voltages:

AVSCDC is connected to analog ground.

AVDCDC= 3.3V (typ).

power consumption

mode	power consumption	Unit
Sleep mode	1200	uW
Playback stereo audio DAC only (capacitor coupled load configuration)	8.3	mW
Record mic stereo input only (audio ADC)	10.3	mW
Record mic mono input only (audio ADC)	6.3	mW
Bypass path (audio stereo line in to HP)	7.7	mW

Current value is at AVDCDC = 3.3 V.

Chip pin Name	MAX Current across I/O @ AVDCDC = 3.3 V
AVDCDC	< 200 mA in normal working mode
AVSCDC	< 20 mA in normal working mode
AVSHP	< 160 mA in normal working mode
	< 1400 mA in case of short circuit
VCAP	< 2 mA in normal working mode
AIP1,AIN1	< 2 mA in normal working mode
AIP2,AIN2	< 2 mA in normal working mode
AIP3,AIN3	< 2 mA in normal working mode
AIP4,AIN4	< 2 mA in normal working mode
MICBIAS1	< 5 mA in normal working mode
MICBIAS2	< 5 mA in normal working mode
AOHPL	< 80 mA in normal working mode
	< 1200 mA in case of short circuit
AOHPR	< 80 mA in normal working mode
	< 1200 mA in case of short circuit
AOLOP,AOLON	< 1 mA in normal working mode
HPSENSE	< 1 mA in normal working mode

The current in case of short circuit is the max value. This current is only sink or drawn until the short circuit detection system acts.

Please refer to Chip Datasheet for more details.

14.7 CODEC internal Registers

Register Name	Function	Address	Reset value
SR	Status Register	00000000 / 0x00 / 00	-
SR2	Status Register 2	00000001 / 0x01 / 01	-
MR	Mode status register	00000111 / 0x07 / 07	-
AICR_DAC	DAC Audio Interface Control Register	00001000 / 0x08 / 08	hD3
AICR_ADC	ADC Audio Interface Control Register	00001001 / 0x09 / 09	hD3
CR_LO	Differential line-out Control Register	00001011 / 0xB / 11	h90
CR_HP	HeadPhone Control Register	00001101 / 0xD / 13	hB0
CR_DMIC	Digital Microphone register	00010000 / 0x10 / 16	h00
CR_MIC1	Microphone1 Control register	00010001 / 0x11 / 17	hB0
CR_MIC2	Microphone2 Control register	00010010 / 0x12 / 18	h30
CR_LI1	Control Register for line1 inputs (for bypass function)	00010011 / 0x13 / 19	h10
CR_LI2	Control Register for line2 inputs (for bypass function)	00010100 / 0x14 / 20	h10
CR_DAC	DAC Control Register	00010111 / 0x17 / 23	h90
CR_ADC	ADC Control Register	00011000 / 0x18 / 24	h90
CR_MIX	Digital Mixer Control Register	00011001 / 0x19 / 25	h00
DR_MIX	Digital Mixer Data Register	00011010 / 0x1A / 26	h00
CR_VIC	Control Register for the ViC	00011011 / 0x1B / 27	h03
CR_CK	Clock Control Register	00011100 / 0x1C / 28	h00
FCR_DAC	DAC Frequency Control Register	00011101 / 0x1D / 29	h00
FCR_ADC	ADC Frequency Control Register	00100000 / 0x20 / 32	h00
CR_TIMER_MSB	MSB of programmable counter	00100001 / 0x21 / 33	h00
CR_TIMER_LSB	LSB of programmable counter	00100010 / 0x22 / 34	h00
ICR	Interrupt Control Register	00100011 / 0x23 / 35	h00
IMR	Interrupt Mask Register	00100100 / 0x24 / 36	hFF
IFR	Interrupt Flag Register	00100101 / 0x25 / 37	-
IMR2	Interrupt Mask Register 2	00100110 / 0x26 / 38	hFF
IFR2	Interrupt Flag Register 2	00100111 / 0x27 / 39	-
GCR_HPL	Left channel headphone Control Gain Register	00101000 / 0x28 / 40	h06
GCR_HPR	Right channel headphone Control Gain Register	00101001 / 0x29 / 41	h06
GCR_LIBYL	Left channel bypass line Control Gain Register	00101010 / 0x2A / 42	h06
GCR_LIBYR	Right channel bypass line Control Gain Register	00101011 / 0x2B / 43	h06
GCR_DAACL	Left channel DAC Gain Control Register	00101100 / 0x2C /	h00

		44	
GCR_DACR	Right channel DAC Gain Control Register	00101101 / 0x2D / 45	h00
GCR_MIC1	Microphone 1 Gain Control Register	00101110 / 0x2E / 46	h00
GCR_MIC2	Microphone 2 Gain Control Register	00101111 / 0x2F / 47	h00
GCR_ADCL	Left ADC Gain Control Register	00110000 / 0x30 / 48	h00
GCR_ADCR	Right ADC Gain Control Register	00110001 / 0x31 / 49	h00
GCR_MIXDACL	DAC Digital Mixer Control Register	00110100 / 0x34 / 52	h00
GCR_MIXDACR	DAC Digital Mixer Control Register	00110101 / 0x35 / 53	h00
GCR_MIXADCL	ADC Digital Mixer Control Register	00110110 / 0x36 / 54	h00
GCR_MIXADCR	ADC Digital Mixer Control Register	00110111 / 0x37 / 55	h00
CR_ADC_AGC	Automatic Gain Control Register	00111010 / 0x3A / 58	h00
DR_ADC_AGC	Automatic Gain Control Data Register	00111011 / 0x3B / 59	h00

14.7.1 CODEC internal registers

14.7.1.1 SR: Status Register

Register Name: SR				Register Address: 0x0			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
PON_ACK	IRQ_ACK	JACK	DAC_LOCKED	Reserved			

Bits	Field	Description
7	PON_ACK	Acknowledge status bit after power on. Read 0: reset value 1: codec is ready to operate
6	IRQ_ACK	Acknowledge status bit after IRQ sending. Read 0: reset value 1: codec has requested an interrupt (IRQ signal activated)
5	JACK	Output Jack plug detection status. Read 0: no jack 1: output jack present
4	DAC_LOCKED	Acknowledge status bit after Sample Rate Converter system is locked Read 0: SRC is not locked. Data from the audio interface are automatically muted. 1: The SRC is locked and operating normally.
3:0	Reserved	Writing has no effect, read as zero.

14.7.1.2 SR2: Status Register 2

Register Name: SR2				Register Address: 0x1			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
Reserved			DAC_UNKNOWN_FS	Reserved			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4	DAC_UNKOWN_FS	Read 0: Measured mean frequency of incoming audio data is within supported values 1: Measured mean frequency of incoming audio data is not within supported values
3:0	Reserved	Writing has no effect, read as zero.

14.7.1.3 MR: Status Register

Register Name: MR				Register Address: 0x7			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
Reserved		ADC_MODE_UPD	ADC_MUTE_STATE	DAC_MODE_UPD	DAC_MUTE_STATE		

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5	ADC_MODE_UPD	Read 0: ADC operation mode is the programmed mode 1: detection of an operation mode update on the ADC
4:3	JADC_MUTE_STATE	Read 00 : ADC not muted 01 : ADC being muted 10 : ADC leaving mute mode 11 : ADC in mute mode
2	DAC_MODE_UPD	Read 0: DAC operation mode is the programmed mode 1: detection of an operation mode update on the DAC
1:0	DAC_MUTE_STATE	Read 00 : DAC not muted 01 : DAC being muted 10 : DAC leaving mute mode 11 : DAC in mute mode

14.7.1.4 AICR_DAC: Audio Interface Control Register

Register Name: AICR_DAC				Register Address: 0x8			
bit7-RW-1	bit6-RW-1	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-1	bit0-RW-1
DAC_ADWL		DAC_SLAVE	SB_AICR_DAC	Reserved		DAC_AUDIOIF	

Bits	Field	Description
7:6	DAC_ADWL	Audio Data Word Length for DAC path. Read / Write 00: 16-bit word length data 01: 18-bit word length data 10: 20-bit word length data 11: 24-bit word length data
5	SB_AICR_DAC	Read/Write: 0 = Audio interface in master mode 1 = Audio interface in slave mode
4	SB_AICR_DAC	Read/Write: 0 = DAC audio interface active 1 = DAC audio interface in power-down mode
3:2	Reserved	Writing has no effect, read as zero.
1:0	DAC_AUDIOIF	Audio Interface mode selection for DAC Read/Write: 00 = Parallel interface 01 = Left-justified interface 10 = DSP interface 11 = I2S interface

NOTES:

- 1 DAC_AUDIOIF should be configured to 11

14.7.1.5 AICR_ADC: Audio Interface Control Register

Register Name: AICR_ADC				Register Address: 0x9			
bit7-RW-1	bit6-RW-1	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-1	bit0-RW-1
ADC_ADWL		Reserved	SB_AICR_ADC	Reserved		ADC_AUDIOIF	

Bits	Field	Description
7:6	ADC_ADWL	Audio Data Word Length for ADC path. Read / Write 00: 16-bit word length data 01: 18-bit word length data

		10: 20-bit word length data 11: 24-bit word length data
5	Reserved	Writing has no effect, read as zero.
4	SB_AICR_ADC	Read/Write: 0 = ADC audio interface active 1 = ADC audio interface in power-down mode
3:2	Reserved	Writing has no effect, read as zero.
1:0	ADC_AUDIOIF	Audio Interface mode selection for ADC Read/Write: 00 = Parallel interface 01 = I2S interface 10 = DSP interface 11 = I2S interface

NOTES:

- 1 ADC_AUDIOIF should be configured to 11

14.7.1.6 CR_LO: differential line-out Control Register

Register Name: CR_LO				Register Address: 0xB			
bit7-RW-1	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LO_MUTE	Reserved		SB_LO	Reserved	LO_SEL		

Bits	Field	Description
7	LO_MUTE	differential line out mute mode. Read/Write 0: mute inactive, Signal applied to line output 1: no signal on line output
6:5	Reserved	Writing has no effect, read as zero.
4	SB_LO	differential line out conditioning circuitry power-down mode. Read/Write 0: active 1: power-down
3:2	Reserved	Writing has no effect, read as zero.
1:0	LO_SEL	differential line-output Amplifier input selection Read/Write 000: DACL 001: DACR 010: DACL + DACR 011: reserved for future use 100: AILatt

		101: AIRatt 110: AILatt + AIRatt 111: reserved for future use
--	--	---

14.7.1.7 CR_HP: HeadPhone Control Register

Register Name: CR_HP				Register Address: 0xD			
bit7-RW-1	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
HP_MUTE	Reserved		SB_HP	Reserved	HP_SEL		

Bits	Field	Description																					
7	HP_MUTE	HeadPhone output signal disabled. Read/Write 0: Signal applied to headphone outputs 1: no signal on headphone outputs, acts as a mute signal																					
6:5	Reserved	Writing has no effect, read as zero.																					
4	SB_HP	headphone output stage power-down mode. Read/Write 0: headphone output stage is active 1: power-down																					
3	Reserved	Writing has no effect, read as zero.																					
2:0	HP_SEL	Headphone Output Amplifier input selection Read/Write <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>HP_SEL</th> <th>HPL input</th> <th>HPR input</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>DACL</td> <td>DACR</td> </tr> <tr> <td>001</td> <td>DACL</td> <td>DACL</td> </tr> <tr> <td>010</td> <td>AILatt</td> <td>AIRatt</td> </tr> <tr> <td>011</td> <td>AILatt</td> <td>AILatt</td> </tr> <tr> <td>100</td> <td>AIRatt</td> <td>AIRatt</td> </tr> <tr> <td>others</td> <td>reserved</td> <td>reserved</td> </tr> </tbody> </table>	HP_SEL	HPL input	HPR input	000	DACL	DACR	001	DACL	DACL	010	AILatt	AIRatt	011	AILatt	AILatt	100	AIRatt	AIRatt	others	reserved	reserved
HP_SEL	HPL input	HPR input																					
000	DACL	DACR																					
001	DACL	DACL																					
010	AILatt	AIRatt																					
011	AILatt	AILatt																					
100	AIRatt	AIRatt																					
others	reserved	reserved																					

14.7.1.8 CR_DMIC: Digital Microphone Control Register

Register Name: CR_DMIC				Register Address: 0x10			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
DMIC_CLKON	Reserved						

Bits	Field	Description
7	DMIC_CLKON	Digital microphone clock (DMIC_CLK) enable

		Read/Write 0= clock off 1= clock on, clock frequency varies with DMIC_RATE and MCLK
6:0	Reserved	Writing has no effect, read as zero.

14.7.1.9 CR_MIC1: Control Register for microphone1 inputs

Register Name: CR_MIC1				Register Address: 0x11			
bit7-RW-1	bit6-RW-0	bit5-RW-1	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
MICSTEREO	MICIDFF1	SB_MICBIAS1	SB_MIC1	Reserved			MIC1_SEL

Bits	Field	Description
7	MICSTEREO	Microphone input mode selection Read/Write 0 = Microphone mono inputs 1 = Microphone stereo inputs This signal affects IN_SEL
6	MICIDFF1	Microphone 1 input mode selection Read/Write 0= single-ended input 1= differential input
5	SB_MICBIAS1	Microphone 1 biasing buffer power-down Read/Write 0 = active 1 = power-down
4	SB_MIC1	Microphone 1 power-down Read/Write 0 = active 1 = power-down
3:1	Reserved	Writing has no effect, read as zero.
0	MIC1_SEL	Microphone 1 input selection Read/Write 0 = selection of AIP1/AIN1 1 = selction of AIP2

14.7.1.10 CR_MIC2: Control Register for microphone2 inputs

Register Name: CR_MIC2				Register Address: 0x12			
bit7-RW-0	bit6-RW-0	bit5-RW-1	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0

Reserved	MICIDFF2	SB_MICBIAS2	SB_MIC2	Reserved	MIC2_SEL
----------	----------	-------------	---------	----------	----------

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	MICIDFF2	Microphone 2 input mode selection Read/Write 0= single-ended input 1= Reserved
5	SB_MICBIAS2	Microphone 2 biasing buffer power-down Read/Write 0 = active 1 = power-down
4	SB_MIC2	Microphone 2 power-down Read/Write 0 = active 1 = power-down
3:1	Reserved	Writing has no effect, read as zero.
0	MIC2_SEL	Microphone 2 input selection Read/Write 0 = selection of AIP3 1 = Reserved

14.7.1.11 CR_LI1: Control Register for line1 inputs (for bypass function)

Register Name: CR_LI1				Register Address: 0x13			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-1
Reserved	LIN1DIFF	Reserved	SB_LIBY1	Reserved		LI1_SEL	

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	LIN1DIF	line 1 input mode selection Read/Write 0 = single-ended input 1 = differential input
5	Reserved	Writing has no effect, read as zero.
4	SB_LIBY1	Line 1 input power-down Read/Write 0 = active 1 = power-down
3:1	Reserved	Writing has no effect, read as zero.
0	LI1_SEL	Line 1 input selection

		Read/Write 0= selection of AIP1/AIN1 1= selection of AIP2
--	--	---

14.7.1.12 CR_LI2: Control Register for line2 inputs (for bypass function)

Register Name: CR_LI2				Register Address: 0x14			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-1
Reserved	LIN2DIFF	Reserved	SB_LIBY2	Reserved			LI2_SEL

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	LIN2DIF	line 2 input mode selection Read/Write 0 = single-ended input 1 = Reserved
5	Reserved	Writing has no effect, read as zero.
4	SB_LIBY2	Line 2 input power-down Read/Write 0 = active 1 = power-down
3:1	Reserved	Writing has no effect, read as zero.
0	LI2_SEL	Line 2 input selection Read/Write 0= selection of AIP3 1= Reserved

14.7.1.13 CR_DAC: Control Register for DAC

Register Name: CR_DAC				Register Address: 0x17			
bit7-RW-1	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
DAC_MUTE	Reserved	DAC_LEFT_ONLY	SB_DAC	Reserved			

Bits	Field	Description
7	DAC_MUTE	DAC soft mute mode. Read/Write 0: mute inactive, digital input signal transmitted to the DAC 1: puts the DAC in soft mute mode
6	Reserved	Writing has no effect, read as zero.

5	DAC_LEFT_ONLY	Left data only are considered. Read/Write 0: DAC right channel active 1: DAC left data are used for left and right channel To avoid any audible pop, it is required to put the DAC in soft mute mode before modifying the DAC_LEFT_ONLY bit.
4	SB_DAC	DAC power-down mode. Read/Write 0: active 1: power-down
3:0	Reserved	Writing has no effect, read as zero.

14.7.1.14 CR_ADC: Control Register for ADC

Register Name: CR_ADC				Register Address: 0x18			
bit7-RW-1	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
ADC_MUTE	ADC_DMIC_SEL	ADC_LEFT_ONLY	SB_ADC	Reserved		ADC_IN_SEL	

Bits	Field	Description
7	ADC_MUTE	ADC soft mute mode Read/Write 0 = mute inactive 1 = puts the ADC in soft mute mode
6	ADC_DMIC_SEL	digital filter input selection Read/Write 0= ADC 1= Digital microphone
5	ADC_LEFT_ONLY	Deactivation of ADC right channel Read/Write: 0= ADC right channel active 1= ADC right channel inactive
4	SB_ADC	ADC power down mode. Read/Write 0: active 1: power-down
3:2	Reserved	Writing has no effect, read as zero.
1:0	ADC_IN_SEL	selection of the signal converted by the ADC Read/Write: If MICSTEREO = 0 00 = Analog input 1 input to left and right channels (ViC automatically considers that ADC_LEFT_ONLY equals '1' to optimize power

		<p>consumption)</p> <p>01 = Analog input 2 input to left and right channels (ViC automatically considers that ADC_LEFT_ONLY equals '1' to optimize power consumption)</p> <p>10 = Reserved for further use</p> <p>11 = Reserved for further use</p> <p>Note that when digital microphone is selected, the ADC_LEFT_ONLY is not automatically modified.</p> <p>If MICSTEREO = 1</p> <p>00 = Analog input 1 input to left channel and Analog input 2 input to right channel</p> <p>01 = Analog input 2 input to left channel and Analog input 1 input to right channel</p> <p>10 = Reserved for further use</p> <p>11 = Reserved for test</p>
--	--	---

NOTE: When ADC right channel is deactivated, right channel output is set to 0

14.7.1.15 CR_MIX: Control Register for digital mixer

Register Name: CR_MIX				Register Address: 0x19			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
MIX_EN	MIX_LOAD	MIX_ADD					

Bits	Field	Description										
7	MIX_EN	digital mixer enable Read/Write 0: digital mixer off 1: digital mixer enabled										
6	MIX_LOAD	Read/Write Refer to DR_MIX for MIX_LOAD description										
1:0	DAC_MIX	Digital mixer control registers address Read/Write <table border="1" data-bbox="550 1608 1018 1868"> <thead> <tr> <th>MIX_ADD</th> <th>Corresponding control register</th> </tr> </thead> <tbody> <tr> <td>000000</td> <td>MIX_0</td> </tr> <tr> <td>000001</td> <td>MIX_1</td> </tr> <tr> <td>000010</td> <td>MIX_2</td> </tr> <tr> <td>000011</td> <td>MIX_3</td> </tr> </tbody> </table>	MIX_ADD	Corresponding control register	000000	MIX_0	000001	MIX_1	000010	MIX_2	000011	MIX_3
MIX_ADD	Corresponding control register											
000000	MIX_0											
000001	MIX_1											
000010	MIX_2											
000011	MIX_3											

14.7.1.16 DR_MIX: Data Register for digital mixer

Register Name: CR_MIX				Register Address: 0x1A			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
MIX_DATA							

Bits	Field	Description
7:0	MIX_DATA	<ul style="list-style-type: none"> Instructions for writing into a register The data must firstly be written into the DR_MIX register. The register address must then be written in CR_MIX (MIX_ADD[5:0]). The data is written into the register when MIX_LOAD is set to '1'. Instructions for reading the values in a register The address of the register to be read is written in CR_MIX (MIX_ADD[5:0]). The content of the corresponding register appears in DR_MIX when MIX_LOAD is set to '0'.

MIX_0: Digital mixer control register 0

Register Name: CR_MIX				Register Address:			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
AIDACL_SEL		AIDACR_SEL		Reserved			DAC_MIX

Bits	Field	Description
7:4	AIDACX_SEL	DAC input selection Read/Write 00: cross inputs 01: normal inputs 10: mixed inputs 11: 0 inputs
3:1	Reserved	Writing has no effect, read as zero.
0	DAC_MIX	Mixer mode on DAC Path Read/Write 0: Playback DAC only 1: Playback DAC + ADC

Note: AIDACX_SEL should be configured to 01 in normal mode.

MIX_1: Digital mixer control register 1

Register Name: CR_MIX				Register Address:			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
MIXDACL_SEL		MIXDACR_SEL		Reserved			

Bits	Field	Description
7:4	MIXDACX_SEL	Mixer input selection on DAC path

		Read/Write 00: normal inputs 01: cross inputs 10: mixed inputs 11: 0 inputs
3:0	Reserved	Writing has no effect, read as zero.

MIX_2: Digital mixer control register 2

Register Name: MIX_2				Register Address:			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
AIADCL_SEL		AIADCR_SEL		Reserved			MIX_REC

Bits	Field	Description
7:4	AIADCX_SEL	Read/Write 00: normal inputs 01: cross inputs 10: mixed inputs 11: 0 inputs
3:1	Reserved	Writing has no effect, read as zero.
0	MIX_REC	Mixer mode on ADC Path Read/Write 0: Record input only 1: Record input + DAC

MIX_3: Digital mixer control register 3

Register Name: CR_MIX				Register Address:			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
MIXADCL_SEL		MIXADCR_SEL		Reserved			

Bits	Field	Description
7:4	MIXADCX_SEL	Read/Write 00: cross inputs 01: normal inputs 10: mixed inputs 11: 0 inputs
3:0	Reserved	Writing has no effect, read as zero.

Note: MIXADCX_SEL should be configured to 01 in normal mode.

14.7.1.17 CR_VIC: Control Register for the codec

Register Name: CR_VIC				Register Address: 0x1B			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-1	bit0-RW-1
Reserved						SB_SLEEP	SB

Bits	Field	Description
7:2	Reserved	Writing has no effect, read as zero.
1	SB_SLEEP	sleep mode. Read/Write 0: normal mode (active) 1: sleep mode
0	SB	complete power-down mode. Read/Write 0: normal mode (active) 1: complete power-down

14.7.1.18 CR_CK: Clock Control Register

Register Name: CR_CK				Register Address: 0x1C			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			SHUTDOWN_CLOCK	CRYSTAL			

Bits	Field	Description												
7:5	Reserved	Writing has no effect, read as zero.												
4	SHUTDOWN_CLOCK	Read/Write 0= normal mode (MCLK active) 1= MCLK is turned off (stuck to '0')												
3:0	CRYSTAL	Selection of the MCLK frequency Read/Write The sampling frequency value is given in the CRYSTAL[3:0] table <table border="1" data-bbox="544 1621 1278 1886"> <thead> <tr> <th>CRYSTAL[3:0]</th> <th>Master Clock Frequency</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>12 MHz</td> </tr> <tr> <td>0001</td> <td>Reserved for further use</td> </tr> <tr> <td>0010</td> <td>13 MHz</td> </tr> <tr> <td>....</td> <td>Reserved for further use</td> </tr> <tr> <td>1111</td> <td>Reserved for further use</td> </tr> </tbody> </table>	CRYSTAL[3:0]	Master Clock Frequency	0000	12 MHz	0001	Reserved for further use	0010	13 MHz	Reserved for further use	1111	Reserved for further use
CRYSTAL[3:0]	Master Clock Frequency													
0000	12 MHz													
0001	Reserved for further use													
0010	13 MHz													
....	Reserved for further use													
1111	Reserved for further use													

NOTE: CRYSTAL should be configured to **0x0** for setting the internal 12MHz master clock SYS_CLK (default).

14.7.1.19 FCR_DAC: DAC Frequency Control Register

Register Name: FCR_DAC				Register Address: 0x1D			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved				DAC_FREQ			

Bits	Field	Description
7:4	Reserved	Writing has no effect, read as zero.
3:0	DAC_FREQ	Selection of the DAC sampling rate (Fs). Read/Write The sampling frequency value is given in the FREQ table.

NOTE: Please refer to section [Sample frequency: FREQ](#).

14.7.1.20 FCR_ADC: ADC Frequency Control Register

Register Name: FCR_ADC				Register Address: 0x20			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved	ADC_HPF	ADC_WNF		ADC_FREQ			

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	ADC_HPF	ADC High Pass Filter enable. Read/Write 0: inactive 1: enable the ADC High Pass Filter
5:4	ADC_WNF	ADC Wind Noise Filter frequency 00= inactive 01= enables the ADC Wind Noise Filter mode 1 10= enables the ADC Wind Noise Filter mode 2 11= enables the ADC Wind Noise Filter mode 3
3:0	ADC_FREQ	Selection of the ADC sampling rate (Fs). Read/Write The sampling frequency value is given in the FREQ table.

NOTE: Please refer to section [Sample frequency: FREQ](#).

14.7.1.21 CR_TIMER_MSB: counter MSB

Register Name: CR_TIMER_MSB				Register Address: 0x21			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
COUNT[15:8]							

Bits	Field	Description
7:0	COUNT[15:8]	MSB of counter programmable number. The counter programmable number register is split into both registers CR_TIMER_MSB and CR_TIMER_LSB.

14.7.1.22 CR_TIMER_LSB: counter LSB

Register Name: CR_TIMER_LSB				Register Address: 0x22			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
COUNT[7:0]							

Bits	Field	Description
7:0	COUNT[7:0]	LSB of counter programmable number The positive number decreases of '1' every slow clock period (256 MCLK periods) and generates an IRQ (TIMER_END) when going from h0001 to h0000.

14.7.1.23 ICR: Interrupt Control Register

Register Name: ICR				Register Address: 0x23			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
INT_FORM		Reserved					

Bits	Field	Description
7:6	INT_FORM	Waveform and polarity of the IRQ signal Read/Write 00: The generated IRQ is a high level 01: The generated IRQ is a low level 10: The generated IRQ is a high level pulse with an 8 MC_CLK cycles duration when using 8-bit parallel control interface or 8 MCLK cycles duration when using SMB control interface 11: The generated IRQ is a low level pulse with an 8 MC_CLK cycles duration when using 8-bit parallel control interface or 8 MCLK cycles duration when using SMB control interface
5:0	Reserved	Writing has no effect, read as zero.

NOTE: INT_FORM should be configured to 00

14.7.1.24 IMR: Interrupt Mask Register

Register Name: IMR				Register Address: 0x24			
bit7-RW-1	bit6-RW-1	bit5-RW-1	bit4-RW-1	bit3-RW-1	bit2-RW-1	bit1-RW-1	bit0-RW-1
LOCK_MASK	SCLR_MASK	JACK_MASK	Reserved		ADC_MUTE_MASK	DAC_MODE_MASK	DAC_MUTE_MASK

Bits	Field	Description
7	LOCK_MASK	Mask for the LOCK_EVENT flag Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)
6	SCLR_MASK	Mask for the SCLR flag. Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)
5	JACK_MASK	Mask for the JACK_EVENT flag. 0: interrupt enabled 1: interrupt masked (no IRQ generation)
4:3	Reserved	Writing has no effect, read as zero.
2	ADC_MUTE_MASK	Mask for the ADC_MUTE_EVENT flag Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)
1	DAC_MODE_MASK	Mask for the DAC_MODE_EVENT flag Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)
0	DAC_MUTE_MASK	Mask for the DAC_MUTE_EVENT flag Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)

NOTES:

- 1 When an interrupt is masked, the event do not generates any change on the IRQ signal, but the corresponding flag value is set to '1' in the IFR register.
- 2 When the IRQ signal is active on level, the IRQ signal is set to the inactive level while the bits IFR[7:0] & (!IMR[7:0]) equals '0'.

- 3 When the IRQ signal is a pulse, the IRQ signal is set to the inactive state until a new non-masked event occurs in IFR[7:0] or until a masked event is unmasked.
- 4 If the 8-bit parallel control interface is selected, MC_CLK must not be stopped in order to propagate IRQ signal.

14.7.1.25 IFR: Interrupt Flag Register

Register Name: IFR				Register Address: 0x25			
bit7-RW-0	bit6-R-0	bit5-RW-0	bit4-R-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LOCK_EVENT	SCLR	JACK_EVENT	Reserved		RDO	GUP	GDO

Bits	Field	Description
7	LOCK_EVENT	Lock status Read 0 = no event 1 = event detected Write 0 = no effect 1 = Reset of the flag
6	SCLR	Left or Right Output short circuit detection status. Read 0 : no event 1 : event detected (due to JACK flag change to '0' or '1') Write 1 to Reset of the flag.
5	JACK_EVENT	Event on output Jack plug detection status. Read 0: no event 1: event detected (due to JACK flag change to '0' or '1'). Write 1 to Reset of the flag.
4:3	Reserved	Writing has no effect, read as zero.
2	ADC_MUTE_EVENT	Read 0 = no event 1 = indicates the beginning or the end of a soft mute on ADC data Write 0 = no effect 1 = Reset of the flag
1	DAC_MODE_EVENT	Read 0 = no event 1 = event detected (due to DAC_MODE_UPD flag change to '0' or '1') Write 0 = no effect 1 = Reset of the flag

0	DAC_MUTE_EVENT	Read 0 = no event 1 = indicates the beginning or the end of a soft mute on DAC data Write 0 = no effect 1 = Reset of the flag
---	----------------	--

NOTE: The flags RUP, RDO, GUP and GDO can be changed after 4 cycles of SYS_CLK from codec being reset.

14.7.1.26 IMR2: Interrupt Mask Register 2

Register Name: IMR2				Register Address: 0x26			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			TIMER_END_MASK	Reserved			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4	TIMER_END_MASK	Mask for the TIMER_END flag Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)
3:0	Reserved	Writing has no effect, read as zero.

14.7.1.27 IFR2: Interrupt Mask Register 2

Register Name: IFR2				Register Address: 0x27			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			TIMER_END	Reserved			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4	TIMER_END	Timer status Read 0 = no event 1 = event detected Write 0 = no effect 1 = reset of the flag
3:0	Reserved	Writing has no effect, read as zero.

14.7.1.28 GCR_HPL: left channel headphone Gain Control Register

Register Name: GCR_HPL				Register Address: 0x28			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-0
LRGO	Reserved		GOL				

Bits	Field	Description
7	LRGO	HP amplifier gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6:5	Reserved	Writing has no effect, read as zero.
4:0	GOL	Left channel HP amplifier gain programming value.

NOTE: Please refer to section "[Programmable attenuation: GO](#)" for more details.

14.7.1.29 GCR_HPR: right channel headphone Gain Control Register

Register Name: GCR_HPR				Register Address: 0x29			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-0
Reserved			GOR				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	GOR	Right channel HP amplifier gain programming value.

NOTE: Please refer to section "[Programmable attenuation: GO](#)" for more details.

14.7.1.30 GCR_LIBYL: left channel bypass line Gain Control Register

Register Name: GCR_LIBYL				Register Address: 0x2A			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-0
LRGI	Reserved		GIL				

Bits	Field	Description
7	LRGI	analog bypass gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6:5	Reserved	Writing has no effect, read as zero.

4:0	GIL	Left channel Line in gain programming value.
-----	-----	--

NOTE: Please refer to section [“Programmable Bypass path attenuation: GI”](#) for more details.

14.7.1.31 GCR_LIBYR: right channel bypass line Gain Control Register

Register Name: GCR_LIBYR				Register Address: 0x2B			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-0
Reserved			GIR				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	GIR	Right channel Line in gain programming value.

NOTE: Please refer to section [“Programmable Bypass path attenuation: GI”](#) for more details.

14.7.1.32 GCR_DACL: Left channel DAC Gain Control Register

Register Name: GCR_DACL				Register Address: 0x2C			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LRGOD	Reserved		GODL				

Bits	Field	Description
7	LRGOD	DAC digital gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6:5	Reserved	Writing has no effect, read as zero.
4:0	GODL	Left channel DAC digital gain programming value.

NOTE: Please refer to section [“Programmable digital attenuation: GOD”](#) for more details.

14.7.1.33 GCR_DACR: right channel DAC Gain Control Register

Register Name: GCR_DACR				Register Address: 0x2D			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			GODR				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.

4:0	GODR	Right channel DAC digital gain programming value.
-----	------	---

NOTE: Please refer to section [“Programmable digital attenuation: GOD”](#) for more details.

14.7.1.34 GCR_MIC1: Microphone 1 Gain Control Register

Register Name: GCR_MIC1				Register Address: 0x2E			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved					GIM1		

Bits	Field	Description
7:3	Reserved	Writing has no effect, read as zero.
2:0	GIM1	Microphone 1 boost stage gain programming value.

NOTE: Please refer to section [“Programmable boost gain: GIM”](#).

14.7.1.35 GCR_MIC2: Microphone 2 Gain Control Register

Register Name: GCR_MIC2				Register Address: 0x2F			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved					GIM2		

Bits	Field	Description
7:3	Reserved	Writing has no effect, read as zero.
2:0	GIM2	Microphone 2 boost stage gain programming value.

NOTE: Please refer to section [“Programmable boost gain: GIM”](#).

14.7.1.36 GCR_ADCL: Left channel ADC Gain Control Register

Register Name: GCR_ADCL				Register Address: 0x30			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LRGID	Reserved	GIDL					
Bits	Field	Description					
7	LRGID	ADC digital gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain					
6	Reserved	Writing has no effect, read as zero.					
5:0	GIDL	Left channel ADC digital gain programming value.					

NOTE: Please refer to the section [“Programmable input attenuation amplifier: GID”](#).

14.7.1.37 GCR_ADCR: Right channel ADC Gain Control Register

Register Name: GCR_ADCR				Register Address: 0x31			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved		GIDR					

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5:0	GIDR	Right channel ADC digital gain programming value.

NOTE: Please refer to the section [“Programmable input attenuation amplifier: GID”](#).

14.7.1.38 GCR_MIXDACL: DAC Left Digital Mixer Control Register

Register Name: GCR_MIXDACL				Register Address: 0x34			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LRGOMIX	Reserved		GOMIXL				

Bits	Field	Description
7	GOMIXL	Mixer gain coupling for DAC path Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6:5	Reserved	Writing has no effect, read as zero.
4:0	GOMIXL	Mixer gain for DAC path. Read/Write 00000 : 0dB 00001 : -1dB ...by step of 1dB 11111 : -31dB

14.7.1.39 GCR_MIXDACR: DAC Right Digital Mixer Control Register

Register Name: GCR_MIXDACR				Register Address: 0x35			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			GOMIXR				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	GOMIXR	Mixer gain for DAC path. Read/Write 00000 : 0dB 00001 : -1dB ...by step of 1dB 11111 : -31dB

14.7.1.40 GCR_MIXADCL: ADC Left Digital Mixer Control Register

Register Name: GCR_MIXADCL				Register Address: 0x36			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LRGIMIX	Reserved		GIMIXL				

Bits	Field	Description
7	LRGIMIX	Mixer gain coupling for input path Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6:5	Reserved	Writing has no effect, read as zero.
4:0	GIMIXL	Mixer gain for input path. Read/Write 00000 : 0dB 00001 : -1dB ... by step of 1dB 11111 : -31dB

14.7.1.41 GCR_MIXADCR: ADC Right Digital Mixer Control Register

Register Name: GCR_MIXADCR				Register Address: 0x37			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			GIMIXR				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	GIMIXR	Mixer gain for input path. Read/Write 00000 : 0dB 00001 : -1dB ... by step of 1dB

		11111 : -31dB
--	--	---------------

14.7.1.42 CR_ADC_AGC: Automatic Gain Control Register

Register Name: CR_ADC_AGC				Register Address: 0x3A			
bit7-RW-0	bit6-RW-0	bit5-RW-1	bit4-RW-1	bit3-RW-0	bit2-RW-1	bit1-RW-0	bit0-RW-0
AGC_EN	ADC_AGC_LOAD	ADC_AGC_LOAD					

Bits	Field	Description												
7	AGC_EN	Enable of the AGC system. Read/Write 0 : inactive 1 : enable the automatic level control												
6	ADC_AGC_LOAD	Read/Write Refer to DR_ADC_AGC for ADC_AGC_LOAD description												
5:0	ADC_AGC_ADD	Address of AGC control registers Read/Write: <table border="1" data-bbox="549 983 1222 1290"> <thead> <tr> <th>ADC_AGC_ADD[5:0]</th> <th>Corresponding control register</th> </tr> </thead> <tbody> <tr> <td>000000</td> <td>ADC_AGC_0</td> </tr> <tr> <td>000001</td> <td>ADC_AGC_1</td> </tr> <tr> <td>000010</td> <td>ADC_AGC_2</td> </tr> <tr> <td>000011</td> <td>ADC_AGC_3</td> </tr> <tr> <td>000100</td> <td>ADC_AGC_4</td> </tr> </tbody> </table>	ADC_AGC_ADD[5:0]	Corresponding control register	000000	ADC_AGC_0	000001	ADC_AGC_1	000010	ADC_AGC_2	000011	ADC_AGC_3	000100	ADC_AGC_4
ADC_AGC_ADD[5:0]	Corresponding control register													
000000	ADC_AGC_0													
000001	ADC_AGC_1													
000010	ADC_AGC_2													
000011	ADC_AGC_3													
000100	ADC_AGC_4													

NOTE: Please refer to section [“AGC system guide”](#) for more details.

14.7.1.43 DR_ADC_AGC: Automatic Gain Control Data Register

Register Name: DR_ADC_AGC				Register Address: 0x3B			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-1
ADC_AGC_DATA							

Bits	Field	Description
7:0	ADC_AGC_DATA	data loaded in AGC control registers Read/Write <ul style="list-style-type: none"> Instructions for writing into a register The data must firstly be written into the DR_ADC_AGC register. The register address must then be written in CR_ADC_AGC (ADC_AGC_ADD[5:0]). The data is written into the register when

		<p>ADC_AGC_LOAD is set to '1'.</p> <ul style="list-style-type: none"> Instructions for reading the values of a register <p>The address of the register to be read is written in CR_ADC_AGC (ADC_AGC_ADD[5:0]). The content of the corresponding register appears in DR_ADC_AGC when ADC_AGC_LOAD is set to '0'.</p>
--	--	--

ADC_AGC_0: AGC control register 0

Register Name: ADC_AGC_0				Register Address:			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved	AGC_STEREO	TARGET				Reserved	

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	AGC_STEREO	<p>selection of the AGC system</p> <p>Read/Write</p> <p>0 =same gain applied to Left and Right channel</p> <p>1 = different gains applied to Left and Right channel</p>
5:1	TARGET	<p>Target output level of the ADC</p> <p>Read/Write:</p> <p>0000 = -6 dB</p> <p>0001 = -7.5 dB</p> <p>... by step of 1.5 dB</p> <p>1111 = - 28.5 dB</p>
0	Reserved	Writing has no effect, read as zero.

ADC_AGC_1: Automatic Gain Control Register 2

Register Name: ADC_AGC_1				Register Address:			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
NG_EN	NG_THR			HOLD			

Bits	Field	Description
7	NG_EN	<p>Selection of the Noise Gate system.</p> <p>Read/Write</p> <p>0: inactive</p> <p>1: enables the noise gate system</p>
6:4	NG_THR	<p>Noise Gate Threshold value.</p> <p>Input level (dB) < Noise Gate Level (dB).</p> <p>Read/Write</p> <p>000: -72 dB</p> <p>001: -66 dB</p>

		...by step of 6dB 111: -30 dB
3:0	HOLD	Hold time before starting AGC adjustment to the TARGET value. Read/Write 0000: 0ms 0001: 2 ms 0010: 4 ms ... Time Step x2 1111: 32.768s

NOTE: Please refer to section [“AGC system guide”](#) for more details.

ADC_AGC_2: AGC control register 2

Register Name: ADC_AGC_2				Register Address:			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
ATK				DCY			

Bits	Field	Description
7:4	ATK	Attack Time - Gain Ramp Down. Read/Write 0000: 32 ms 0001: 64 ms ...by step of 32 ms 1111: 512 ms
3:0	DCY	Decay Time - Gain Ramp up. Read/Write 0000: 32 ms 0001: 64 ms ...by step of 32 ms 1111: 512 ms

NOTES:

- 1 DCY and ATK registers values are delays between each step of gain.
- 2 Please refer to section [“AGC system guide”](#) for more details.

ADC_AGC_3: AGC control register 3

Register Name: ADC_AGC_3				Register Address:			
Bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved				AGC_MAX			

Bits	Field	Description
------	-------	-------------

7:5	Reserved	Writing has no effect, read as zero.
4:0	AGC_MAX	Maximum Gain Value to apply to the ADC path.

NOTES:

- 1 Please refer below table for AGC_MAX setup.
- 2 Please refer to section “[AGC system guide](#)” for more details.

AGC_MAX	Gain Value	AGC_MAX	Gain Value	AGC_MAX	Gain Value	AGC_MAX	Gain Value
00000	0	01000	12	10000	23	11000	32
00001	1.5	01001	13.5	10001	23	11001	33.5
00010	3	01010	15	10010	23	11010	35
00011	4.5	01011	16.5	10011	24.5	11011	36.5
00100	6	01100	18	10100	26	11100	38
00101	7.5	01101	19.5	10101	27.5	11101	39.5
00110	9	01110	21	10110	29	11110	41
00111	10.5	01111	22.5	10111	30.5	11111	42.5

ADC_AGC_4: AGC control register 4

Register Name: ADC_AGC_4				Register Address:			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			AGC_MIN				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	AGC_MIN	Maximum Gain Value to apply to the ADC path.

NOTES:

- 1 Please refer to below table for AGC_MIN setup.

AGC_MIN	Gain Value	AGC_MIN	Gain Value	AGC_MIN	Gain Value	AGC_MIN	Gain Value
00000	0	01000	12	10000	23	11000	32
00001	1.5	01001	13.5	10001	23	11001	33.5
00010	3	01010	15	10010	23	11010	35
00011	4.5	01011	16.5	10011	24.5	11011	36.5
00100	6	01100	18	10100	26	11100	38
00101	7.5	01101	19.5	10101	27.5	11101	39.5
00110	9	01110	21	10110	29	11110	41
00111	10.5	01111	22.5	10111	30.5	11111	42.5

- 2 Please refer to section “[AGC system guide](#)” for more details.

14.8 Programmable gains

This section helps you to configure the programmable gain amplifier in the CODEC.

Internal signal VREFP is connected to AVDCDC Pin and internal signal VREFN is connected to AVSCDC Pin.

In this section, VREF equals to (VREFP – VREFN).

14.8.1 Programmable boost gain: GIM

The following table gives the relation between the gain and the input level for the microphone input amplifier when **GID = 0**.

GIM[2]	GIM[1]	GIM[0]	Gain value (dB)	Maximum input amplitude (Vpp differential) (FS)
0	0	0	0	0.85*VREF
0	0	1	4	0.536*VREF
0	1	0	8	0.338*VREF
0	1	1	12	0.213*VREF
1	0	0	16	0.134*VREF
1	0	1	20	0.085*VREF
1	1	0	20	0.085*VREF
1	1	1	20	0.085*VREF

NOTES:

- 1 Analog input amplitude value is not more than 'Maximum input amplitude' for Vpp differential.
- 2 This maximum analog input amplitude is referenced as Full Scale (FS). After conversion, the corresponding digital code of the output value varies from 0x7FFF down to 0x8000 for a 16-bit word. When the analog input amplitude value is greater than FS, the dynamic characteristics are not guaranteed.
- 3 When a change occurs on GIDi inputs, data are valid on the digital output after about 64 sample periods. If the HPF is activated, data are valid after about 64 sample periods but the offset cancellation is not still completed at this time due to its internal time constant.

14.8.2 Programmable input gain amplifier: GID

The digital gain of ADC path may be programmed through the registers bits GIDL and GIDR.

The value of the gain is programmable from 0 to 23dB with a pitch of 1dB.

The gain and input levels are obtained according to the following table:

GID						Decimal decoded value	Gain value (dB)	Maximum input amplitude (Vpp. Differential) (FS)
0	0	0	0	0	0	0	0	0.85*VREF
0	0	0	0	0	1	1	1	0.757*VREF
0	0	0	0	1	0	2	2	0.6021*VREF
...								
x	y	z	t	u	v	i	i	$0.85/\{10^{(i/20)}\} * VREF$
...								
0	1	0	1	1	1	23	23	0.06*VREF
0	1	1	0	0	0	24	24	0.06*VREF
...								
1	0	1	0	1	0	43	43	0.006*VREF

NOTE: The last column of the table shows the maximum analog input to **be applied to** the MICi inputs. The value is given in Vpp differential. These values refer to the external voltage reference VREF **equal to** (VREFP – VREFN). The voltage levels depend on the VREF voltage.

14.8.3 Programmable digital attenuation: GOD

The attenuation of DAC output amplifier may be programmed independently for the both channels through the registers bits GODL and GODR.

The value of the gain GODL/R is programmable from +0 to –31dB with 1 dB pitch. The gain and output levels are obtained according to the following table:

GOD					Decimal decoded value	Gain Value (dB)
0	0	0	0	0	0	0
0	0	0	0	1	1	-1
...						
0	0	1	1	0	6	-6
...						
1	1	1	1	0	30	-30
1	1	1	1	1	31	-31

14.8.4 Programmable attenuation: GO

The attenuation of Headphone output amplifier may be programmed independently for the both

channels through the registers bits GOL and GOR.

The value of the gain GOL/R is programmable from +6 to -25dB with 1 dB pitch. The gain and output levels are obtained according to the following table:

GO					Decimal decoded value	Gain Value (dB)	Maximal PGAT input amplitude (Vpp)	Maximal PGAT output amplitude (Vpp)
0	0	0	0	0	0	+6	$0.425 \cdot V_{REF}$	$0.85 \cdot V_{REF}$
0	0	0	0	1	1	+5	$0.478 \cdot V_{REF}$	$0.85 \cdot V_{REF}$
				
0	0	1	0	1	5	+1	$0.757 \cdot V_{REF}$	$0.85 \cdot V_{REF}$
0	0	1	1	0	6	0	$0.85 \cdot V_{REF}$	$0.85 \cdot V_{REF}$
0	0	1	1	1	7	-1	$0.85 \cdot V_{REF}$	$0.757 \cdot V_{REF}$
				
1	1	1	1	0	30	-24	$0.85 \cdot V_{REF}$	$0.054 \cdot V_{REF}$
1	1	1	1	1	31	-25	$0.85 \cdot V_{REF}$	$0.048 \cdot V_{REF}$

NOTES:

- 1 When headphone driver is loaded by a 16 Ohm load, setting GOL/R = 0 is possible. However, set GOL/R to 9 at the maximum to preserve dynamic performances. The output stage is sized to support a 70mA current at most.
- 2 The last column of the table shows the analog output voltage in the condition of a digital input at FS (Full Scale). The value is given in Vpp single-ended.
- 3 These values refer to the external voltage reference VREF equal to (VREFP – VREFN). The voltage levels depend on the VREF voltage.

14.8.5 Programmable Bypass path attenuation: GI

The analog input gain may be programmed through GIL/R.

The value of the gain is programmable from +6 to -25dB with a pitch of 1dB. The gain and input levels are obtained according to the following table:

GI					Decimal decoded value	Gain value (dB)	Maximum input amplitude (Vpp) (FS)
0	0	0	0	0	0	+6	$0.425 \cdot V_{REF}$
0	0	0	0	1	1	+5	$0.478 \cdot V_{REF}$
0	0	0	1	0	2	+4	$0.536 \cdot V_{REF}$

x	y	z	t	u	i	6 - i	$0.85/\{10^{\{(6 - i)/20\}}\} * V_{REF}$
0	0	1	1	0	6	0	0.85*VREF
					...		0.85*VREF
1	1	1	1	1	31	-25	0.85*VREF

The last column of the table gives the maximum analog input **voltage**. The value is given in Vpp. These values refer to the external voltage reference VREF **equal** to (VREFP – VREFN). The voltage levels depend on the VREF voltage.

14.8.6 Programmable digital mixer gain: GIMIX and GOMIX

The following table shows the relation between the gain and GIMIX/GOMIX.

GIMIX or GOMIX	Gain value (dB)
00000	0
00001	-1
00010	-2
00011	-3
...	...
11101	-29
11110	-30
11111	-31

14.8.7 Gain refresh strategy

GI* and GO* gains are controlled through the control interface. To avoid sound artifacts, the gain increases or decreases each time the gain stage output crosses the zero value. **Tcrossout** time-out counter forces the gain to be updated if a zero crossing event doesn't occur. After each gain step, zero crossing events are ignored during at least **Tcrossmin**.

In case that the gain coupling between both left and right channels is active (LRGi different from RLGi), gain stepping of each channel is independent from the other depending on zero crossing event occurrence.

The duration of Tcrossout and Tcrossmin are given below:

MCLK (MHz)	Tcrossout (ms)	Tcrossmin (ms)
12	21.8	0.171
13	20.2	0.158

14.9 Configuration of the headphone output stage

In cap-coupled connection, codec uses the LOAD register bit to control the ramping duration. Inappropriate setting will lead to a too long or too fast ramping and will create audio artifacts.

To prevent pop-up noise generation due to floating nodes when no load is plugged in the jack connector, it is required to add some resistor devices that act as pull down function (named Rhpl and Rhpr in section “Headphone connection” and section “Required external components”).

Its value has to be determined as following:

Working Mode	Load resistor and bypass capacitor values	LOAD value	Rhpdo value
Driving Headphone	16 Ohm / 220uF	0	470 Ohm typ.
Driving Lineout	10k Ohm / 1uF	1	4.7k Ohm max.

14.10 Out-of-band noise filtering

An internal analog Low Pass Filter at the DAC output is designed to remove the out-of-band noise generated by the delta sigma modulation (Noise Shaper). The internal LPF reduces the amount of energy contained in the wide band part (> 24 kHz) of the output signal. The out-of-band noise, when not removed, can be damageable in some high quality applications.

This filter is always working and does not need configure.

14.11 Output short-circuit protection (headphone output)

Analog short-circuit protection in the output stage has been implemented to prevent excessive current from flowing through AOHPL, AOHPR output pins. This prevents the output stage from over-heating.

The system detects the following **case**:

- Abnormal headphone load.

14.11.1 Indication of the short circuit detection

When such an overload is detected on one of AOHPL, AOHPR output pins,

- An interrupt is sent on the IRQ pin and the SCMC flag in the IFR register is set to ‘1’.
- Internally to codec:
 - Automatic power-down of the 2 output amplifiers (AOHPL, AOHPR signals) when a short-circuit is detected on AOHPL or AOHPR pin (SCMC flag is set to ‘1’).

14.11.2 Reset of short circuit detection

The following sequence has to be to apply:

- 1 Mask the interrupt by writing '1' in the Interrupt Control Register.
- 2 Handle the cause of short-circuit according to the events presented in following paragraphs (Capacitor-coupled headphone connection).
- 3 Update the short-circuit flag by writing '1' in the Interrupt Flag Register.
- 4 Check the reset of flag by reading the Interrupt Flag Register. The bit must be equal to '0'. If it remains at '1', that means that short-circuit is not resolved.
- 5 Enable the interrupt by writing '0' in the Interrupt Control Register.

14.11.3 Capacitor-coupled headphone connection

It is up to the application to put the output stage in power down mode (SB_HP = '1'), to put codec in sleep or complete power-down mode, to reset it.

The short-circuit will be solved by the following events:

- Removal of the inserted jack. (needs the use of HPSENSE pins)
- Reset of codec. (NRST signal)
- Putting the output stage in power-down mode. (SB_HP=1)
- Putting codec in sleep mode. (SB_SLEEP=1)
- Putting codec in complete power-down mode. (SB=1)

14.12 Sampling frequency: FREQ

The sampling frequency value is given in the FREQ table below.

FREQ[3:0]	Sampling Rate (Fs)
0000	8 kHz
0001	11.025 kHz
0010	12 kHz
0011	16 kHz
0100	22.05 kHz
0101	24 kHz
0110	32 kHz
0111	44.1 kHz
1000	48 kHz
1001	88.2 kHz
1010	96 kHz
1011	Reserved for further use
....

1111	Reserved for further use
------	--------------------------

14.13 Programmable data word length

The Data Word Length block (DWL) allows selecting the length of the input data and of the output data between 24-/20-/18-/16-bit to **AICR_DAC.DAC_ADWL** and **AICR_ADC.ADC_ADWL** (respectively for the DAC and ADC paths) in accordance with the following table:

ADWL	Word length
0 0	16-bit word length data
0 1	18-bit word length data
1 0	20-bit word length data
1 1	24-bit word length data

The size of the buses is always 24 bits, but the input/output data only use the number of MSB programmed with ADWL. The LSB are considered as '0' in input and set to '0' in output.

The capability to use a data word length of 16 bits is kept for compatibility with standard applications.

14.14 Ramping system note

An internal mechanism is used to reduce output glitches when the headphone stage enters or leaves the power-down mode.

When the SB_HP is set to '1', the headphone output voltages (AOHPL, AOHPR) are slowly decreased in the same time from $V/2$ down to 0.

When the SB_HP is set to '0', the headphone output voltages (AOHPL, AOHPR) are slowly increased in the same time from 0 to $V_{REFP}/2$.

After power supplies ramp up, the CODEC starts its internal initialization sequence and SR.PON_ACK register is changed when the initialization sequence is complete.

An interrupt request is sent when the ramp completes.

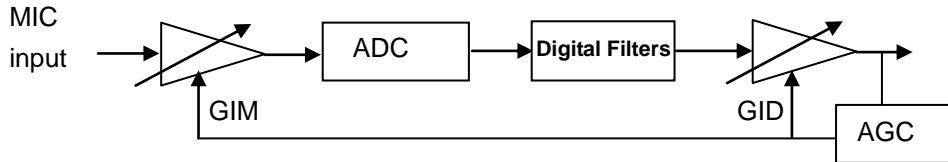
Do not change the level of SB_HP as long as the initialization sequence is not complete. If the change occurs, the success of the **initialization** sequence is not **guaranteed**.

In order to prevent audible glitch, it is required to power-down the output stage (SB_HP=1) before changing the output load with CR1.LOAD.

Please refer to ["Anti-pop operation sequences"](#) for details.

14.15 AGC system guide

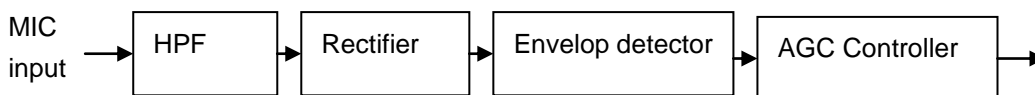
For the microphone input to ADC path, an Automatic Gain Control (AGC) system allows to optimize the signal swing at the input of the ADC.



The AGC circuit compares the output of the ADC to a level and increases or decreases the gain of the microphone preamplifier and the digital gain to compensate. The full dynamic range of the ADC can be used automatically if the audio from the microphone is to be output digitally through the ADC.

The AGC_EN register bit enables the AGC system, in this case IN_SEL must be equal to "00" or "01".

AGC Block Diagram:



The AGC system is used at the MIC input.

The HPF filter characteristics: Cut Frequency =300 Hz.

In the in AGC mode, the system of gain control will directly assign the values of the gains GIDL, GIDR and GIM1 (or GIM2).

14.15.1 AGC operating mode

TARGET sets the desired ADC output range level. The AGC system adapts the gain stages (GID and GIM) in order to best reach this target. AGC_MAX and AGC_MIN fix the limits of the gain variation.

Please refer to ["CODEC Operating modes"](#) for the AGC System diagram in the "CODEC Power Diagram".

In order that the AGC system should not alter the dynamic content of the signal (voice "tonic" for instance) by continuously adapting the gain to fit the target level, the time between two consecutive gain adjustments is modifiable by the HOLD register value.

After this delay:

- If the output level is lower than TARGET, the gain is increased step by step in accordance to the DCY register value.
- If the output level is higher than TARGET, the gain is decreased step by step in accordance to the ATK register value.

The following figure illustrates the behavior of AGC system:

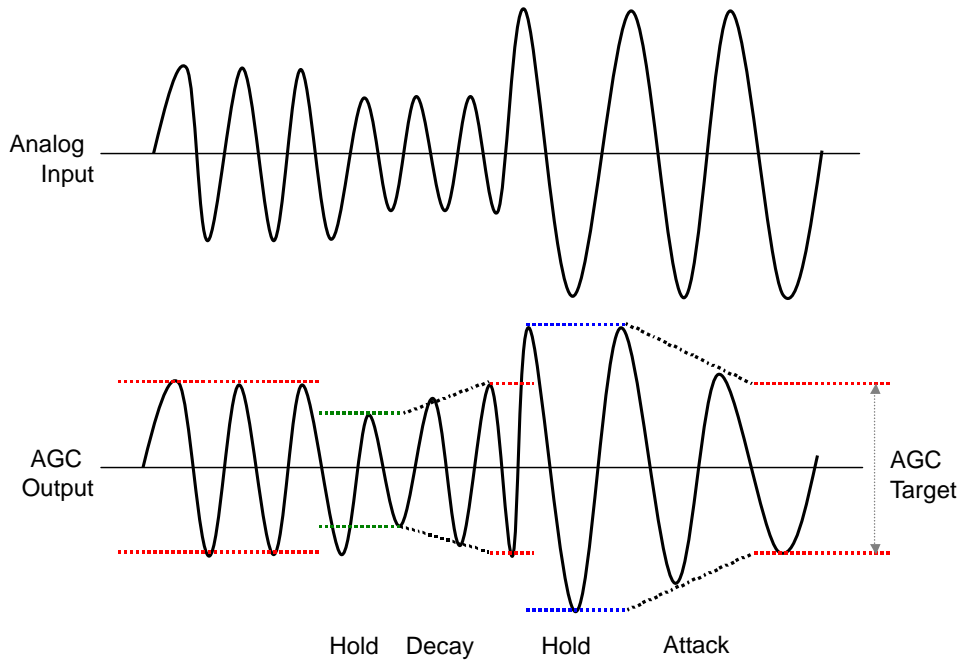


Figure 14-3 AGC adjusting waves

A noise-gating feature, enabled by the NG_EN register bit, prevents gain increases when no signal or small signal is present at the input. The noise gate threshold is set by the NG_THR register value. The following graph shows a more detailed application. The following graph summarizes the operations and shows more details.

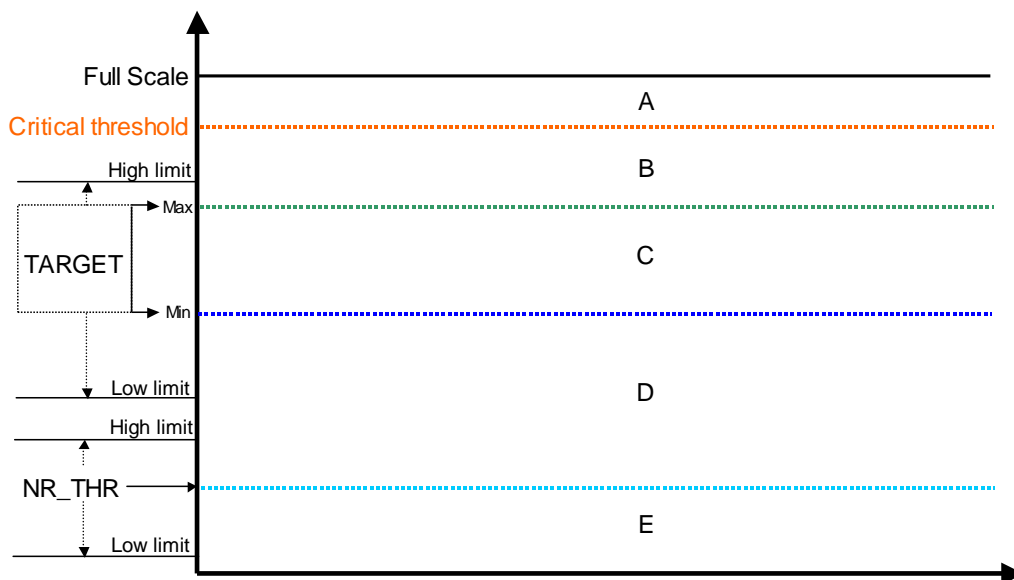
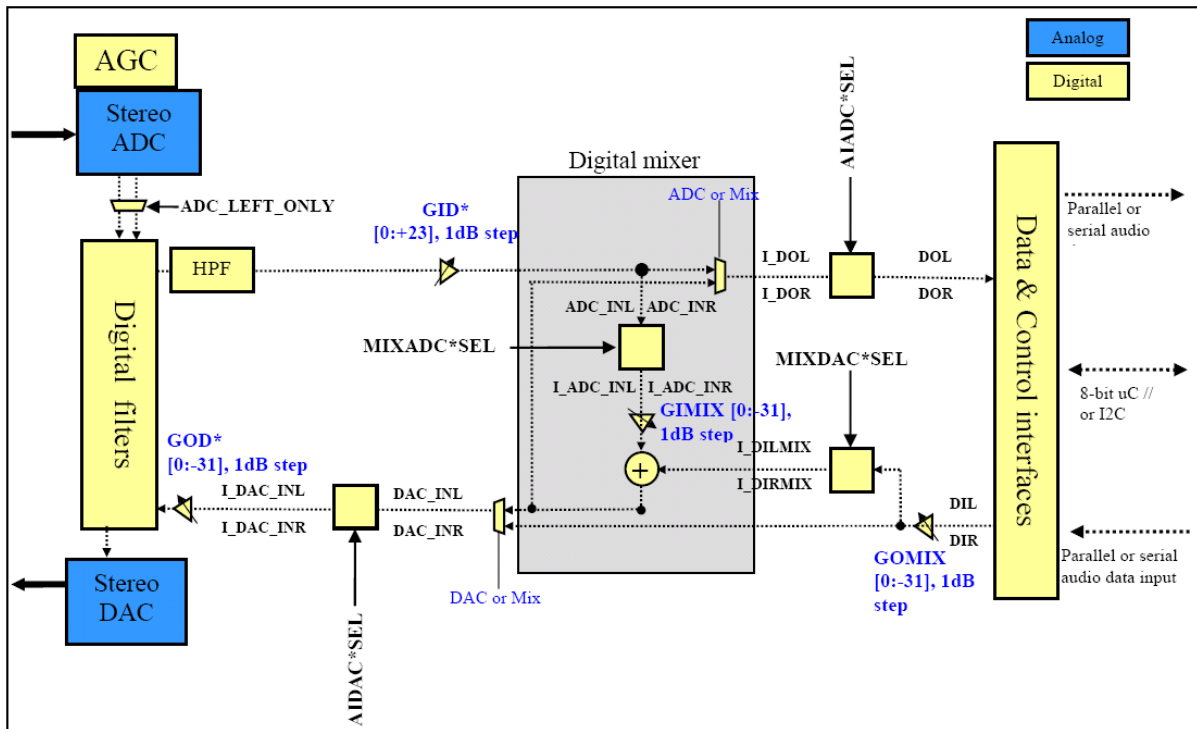


Figure 14-4 AGC adjust areas

The areas from A to E are **different working areas** of AGC system, which is listing below:

- A: If the signal level is in this critical area: the AGC system decreases quickly the gain at the input of the ADC until the signal goes under the critical threshold.
- B: If the signal level remains in this area after the HOLD delay: the AGC system decreases the gain at the input of the ADC until the signal reaches the target area with a slope defined by AGC3.ATK register value.
- C: If the signal level is in this area: the AGC system does not perform gain adjustment.
- D: If the signal level remains in this area after the HOLD delay: the AGC system increases gain at the input of the ADC until the signal reach the target area with a slope defined by AGC3.DCY register value.
- E: If the signal level is in this range: the AGC system considers the signal as noise and does not perform gain adjustment.

14.16 Digital Mixer description



CODEC includes a digital mixer which provides a loopback of the ADC output to the DAC and Headphone output and a loopback of the mixer output to the record path.

Two gains GIMIX[4:0] and GOMIX[4:0] are added to each input of the mixer to adapt the amplitude of the mixed signal. A zero-crossing detection is included on each gain stage to minimize the zipper noise.

When ADC_LEFT_ONLY = '1', the right output of stereo ADC is set to '0' (ADC_INR = '0').

A digital multiplexer allows choosing between the ADC signal and the mixer output signal on the record path.

AIDAC*SEL, AIADC*SEL, MIXDAC*SEL and MIXADC*SEL functionality are applied between the Digital Mixer and the Data interface. The behavior is the following:

AIDACLSEL1	AIDACLSEL0	I_DAC_INL	AIDACRSEL1	AIDACRSEL0	I_DAC_INR
0	0	DAC_INR	0	0	DAC_INL
0	1	DAC_INL	0	1	DAC_INR
1	0	(DAC_INL + DAC_INR)/2	1	0	(DAC_INL + DAC_INR)/2
1	1	0	1	1	0

AIADCLSEL1	AIADCLSEL0	DOL	AIADCRSEL1	AIADCRSEL0	DOR
0	0	I_DOR	0	0	I_DOL
0	1	I_DOL	0	1	I_DOR

1	0	$(I_DOL+I_DOR)/2$	1	0	$(I_DOL+I_DOR)/2$
1	1	0	1	1	0

MIXDACLSEL1	MIXDACLSEL 0	I_DILMIX	MIXDACRSEL1	MIXDACRSEL0	I_DIRMIX
0	0	DIL	0	0	DIR
0	1	DIR	0	1	DIL
1	0	$(DIL+DIR)/2$	1	0	$(DIL+DIR)/2$
1	1	0	1	1	0

MIXADCLSEL1	MIXADCLSEL 0	I_ADC_INL	MIXADCRSEL1	MIXADCRSEL 0	I_ADC_INR
0	0	ADC_INR	0	0	ADC_INL
0	1	ADC_INL	0	1	ADC_INR
1	0	$(ADC_INL+ADC_INR)/2$	1	0	$(ADC_INL+ADC_INR)/2$
1	1	0	1	1	0

14.17 Digital microphone interface

CODEC accepts bitstream from digital microphone and converts it into audio data at the sample rate (F_s) selected in FCR_ADC register. CODEC provides a clock (DMIC_CLK) and receives data on DMIC_IN at the same frequency. DMIC_CLK frequency depends on MCLK frequency selection in CCR register.

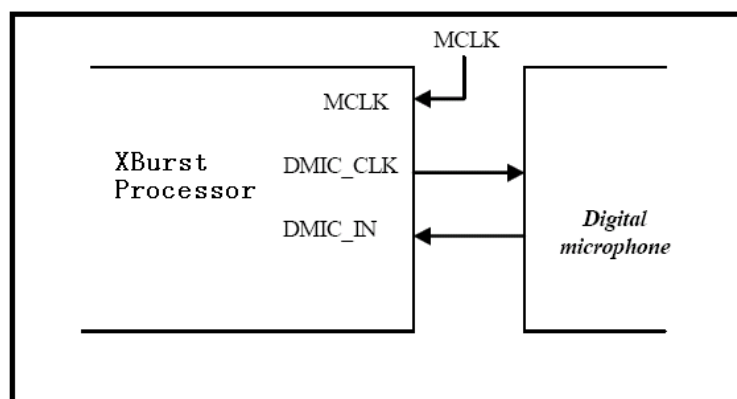


Figure 14-5 Digital microphone interface connection

After conversion, the corresponding digital code of the output value varies from 0x7FFF down to 0x8000 for a 16-bit word, coded in 2's complement.

CODEC can receive simultaneously data from two digital microphones.

14.17.1 Timing Diagram

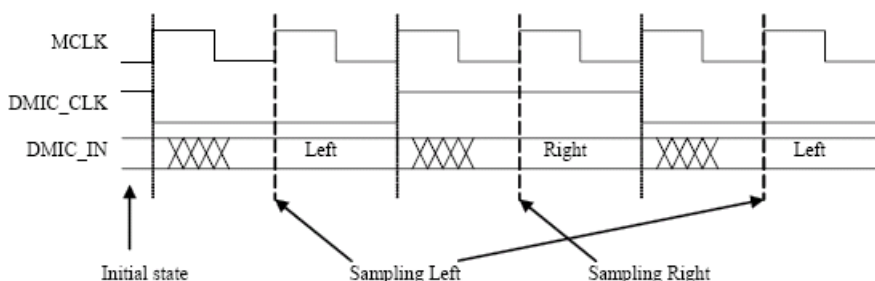
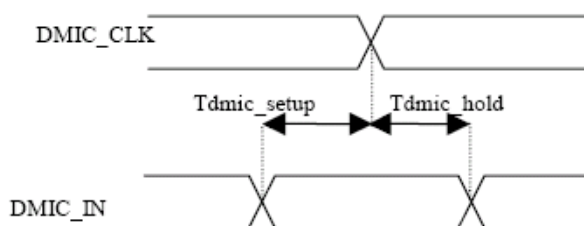


Figure 14-6 Digital microphone timing diagram at MCLK = 12 MHz

(DMIC_CLK = 3 MHz) and MCLK = 13 MHz (DMIC_CLK = 3.25 MHz)

14.17.2 Timings



Parameter	Symbol	Min	Typ	Max	Unit
DMIC_CLK frequency	F_{dmic_clk}	3	-	3.25	MHz
DMIC_CLK duty cycle	D_{dmic_clk}	0.4	0.5	0.6	-
DMIC_IN setup time	T_{dmic_setup}	$T_{MCLK} + 10$	-	-	ns
DMIC_IN hold time	T_{dmic_hold}	0	-	-	ns

14.17.3 Noise template (TBC)

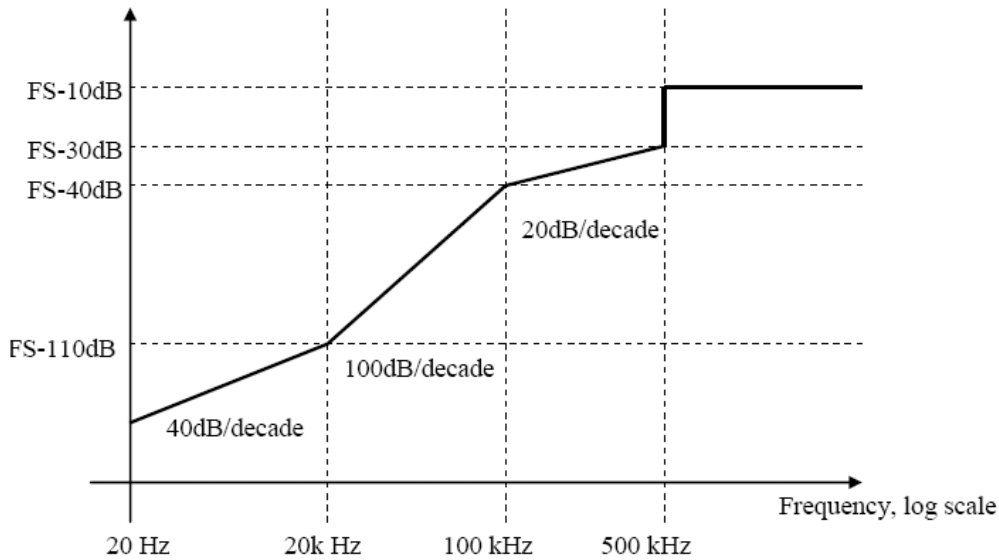


Figure 14-7 Digital microphone modulation noise reference spectrum

(with FFT resolution = 20 Hz and 7 terms Blackman-Harris windowing)

If the noise at the digital microphone output is higher than the reference in the [20 Hz – 20 kHz] bandwidth, the SNR will be limited by the digital microphone in-band noise.

If the noise at the digital microphone output is higher than the reference for frequencies beyond 20 kHz, the SNR will be limited by the aliasing of the digital microphone quantization noise.

14.18 CODEC Operating modes

Different operating modes are available:

- Power-up mode: During power on time, CODEC is in this mode.
- Reset mode: When NRST is low, CODEC is in this mode.
- Soft mute mode: When DAC_MUTE is 1, CODEC is in this mode.
- Complete Power-down mode: After RESET, CODEC is in this mode.
- SLEEP modes: When SB_SLEEP is 1, CODEC is in this mode.
- Normal mode: When CODEC is not in above mode, it is in this mode. This mode has three modes: RECORD mode, REPLAY mode, RECORD_REPLAY mode.

The power diagram is shown below.

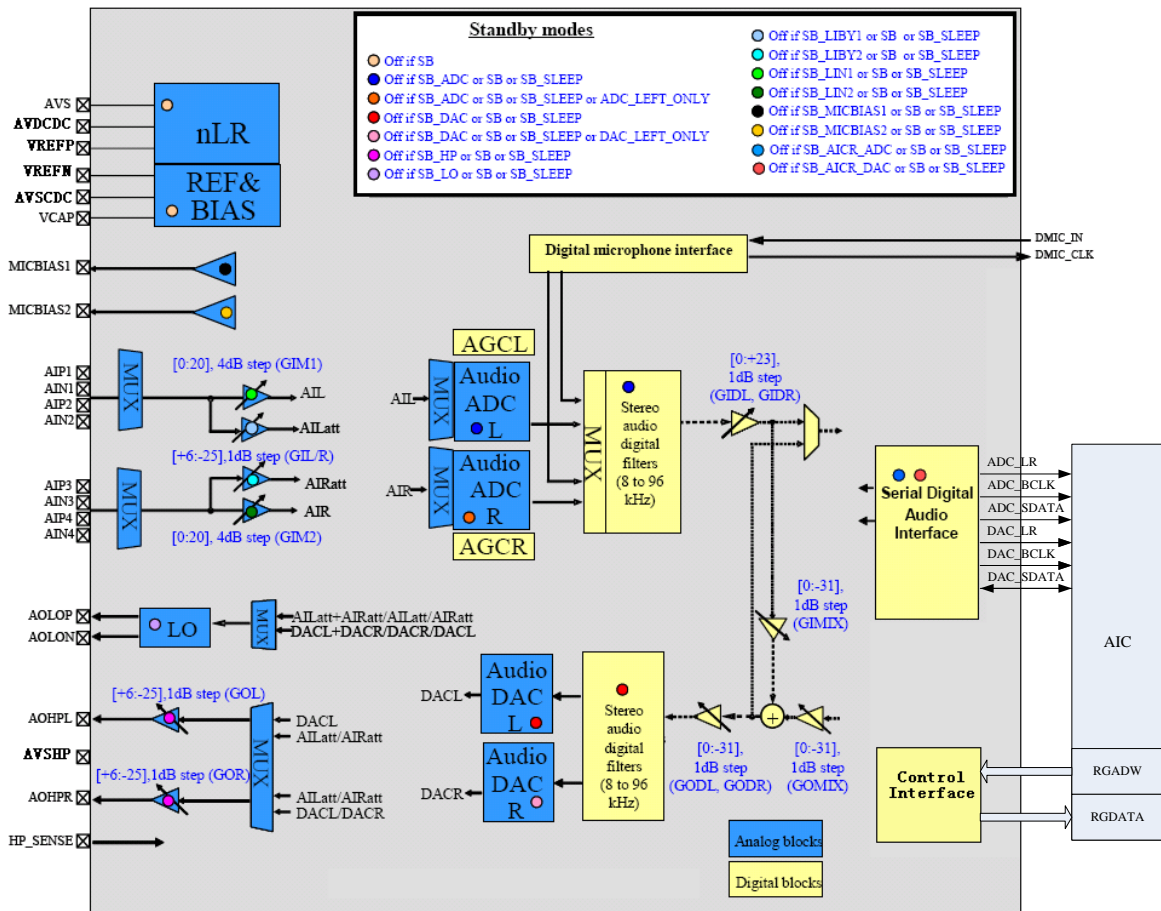


Figure 14-8 CODEC Power Diagram

There are many power parts of CODEC. Any part could be powered down independently.

14.18.1 Power-On mode and Power-Off mode

When the power supply ramps up, CODEC enters the power-on mode. During the reset, the CODEC is put in stand-by in order to reduce audible pops.

The CODEC doesn't handle the power supply ramp down on itself. The software has to turn the CODEC in complete stand-by mode before the power supply starts to ramp down.

14.18.2 RESET mode

The reset input signal is asynchronous; the reset minimum duration is one SYS_CLK cycle. During the power-up mode and system reset, the CODEC goes into Reset mode. After system reset the CODEC will exit Reset mode and go to STANDBY mode.

NOTES:

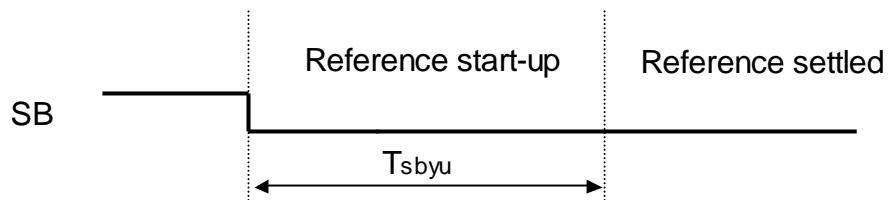
- 1 Except during the power-up mode, do NOT perform any reset in order to avoid audible pops.

- Resetting the CODEC during normal operating mode will turn instantaneously the CODEC in STANDBY mode. This will lead to generate a large audible pop.

14.18.3 STANDBY mode

CODEC goes to STANDBY mode when the SB register bit equals 1, and all functions including ADC path, DAC path and analog references will stop and whole CODEC is shutdown for saving power. CODEC is complete down in this mode.

During the STANDBY mode, the power consumption is reduced to a minimum, so it is also called Complete Power-Down mode. When SB is set to '0', CODEC leaves the STANDBY mode. It is necessary to wait some time before the CODEC references settle. This time is called T_{sbyu} . When CODEC reference settled, it is in SLEEP mode.

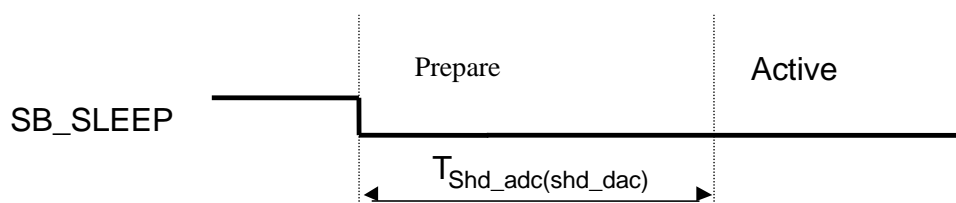


Please refer to the section "[Timing parameters](#)" for the T_{sbyu} Value.

14.18.4 SLEEP mode

When SB_SLEEP is set to 1, CODEC enters in sleep mode. The logical part and the analog functions, except the voltage and biasing references, enter in power-down mode. So, the power consumption is reduced without penalizing the start-up time.

When SB_SLEEP falls, CODEC leaves the corresponding STANDBY mode; it is necessary to wait some time before the CODEC reaches the normal mode. Depending on the selected mode, this time is either called T_{shd_adc} (SB_ADC=0) for the ADC path or T_{shd_dac} (SB_DAC=0) for the DAC path.



Please refer to the section "[Timing parameters](#)" for the T_{shd_adc} and T_{shd_dac} Value.

14.18.5 Initial all the gain

When CODEC turn into STANDBY mode, need initial all the gain:

- Firstly set a gain that different you wished (such as '1') to GIM1 / GIM2 / GIDL / GIDR / GIL / GIR / GOL / GOR / GODL / GODR.
- Secondly wait about 10ms and than set a gain you wished (such as '0') to GIM1 / GIM2 / GIDL / GIDR / GIL / GIR / GOL / GOR / GODL / GODR.

After the step, you can change the gain you wished at any times

14.18.6 Soft Mute mode

Soft Mute mode is used in order to reduce audible parasites when before the DAC enters or after leaves the Normal mode. Set the DAC_MUTE register bit to 1, it will go to Soft Mute mode.

Set DAC_MUTE to 1 puts the DAC in Soft Mute mode. The CODEC decreases progressively the digital gain from 0dB to $-\infty$. When the gain down sequence is completed, the signal of the DAC is equal to 0 whatever the value of the digital input data is. **The CODEC sets IFR.GDO register bit to 1, then generates an interrupt if IMR.GDO_MASK is 0.**

During Soft Mute mode, the DAC is still converting but the output final voltages (**AOLOP, AOHP**) are equal to $V_{REF}/2$, so the differential of the Headphone voltage is zero that cause no sound output.

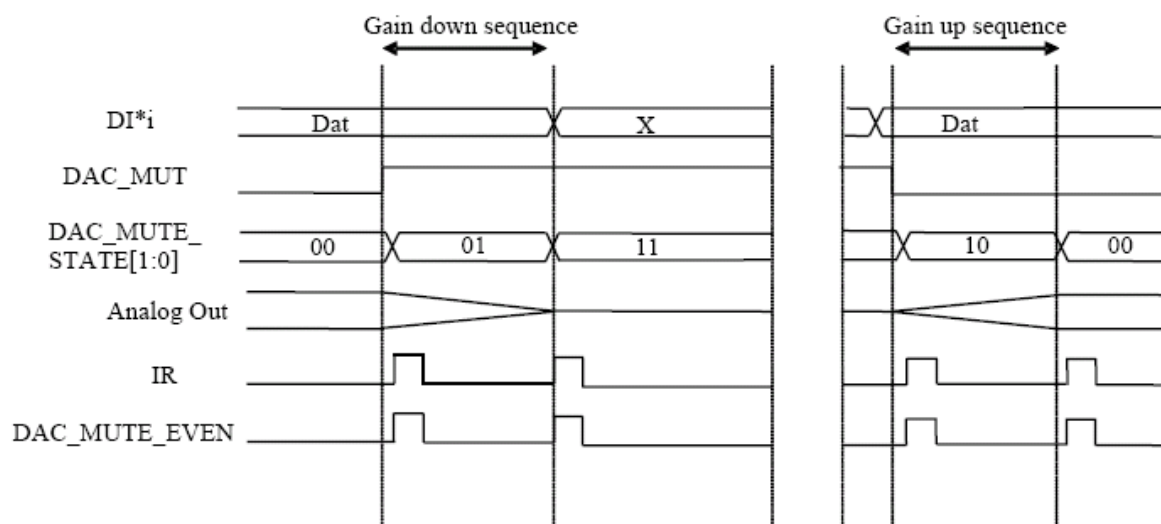


Figure 14-9 DAC Gain up and gain down sequence

In the opposite, when DAC_MUTE is set to 0, the DAC leaves the Soft Mute mode by increasing

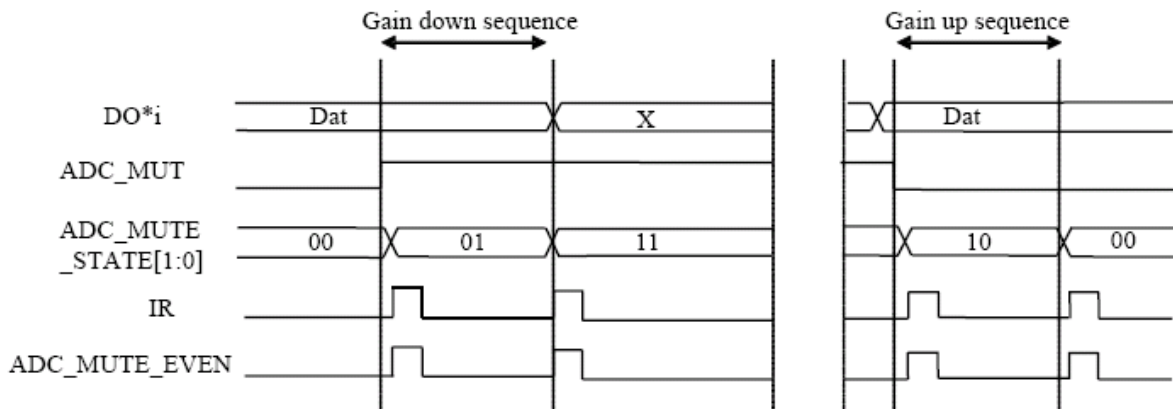


Figure 14-10 ADC Gain up and gain down sequence

progressively the digital gain from $-\infty$ to 0dB. When the gain up sequence is completed, the DAC returns in Normal mode. **The CODEC sets IFR.GUP register bit to 1, then generates an interrupt if IMR.GUP_MASK is 0.**

After exiting Soft Mute mode, the DAC output will flow the DAC input data, and there is sound in the Headphone.

The duration of gain down and gain up sequences are nearly independent of F_s and is about 23ms.

NOTES:

- 1 Do NOT change the value of `DAC_MUTE` while the effect of the previous change is not reached, or the working is not guaranteed.
- 2 Do NOT enter in stand-by mode while the gain sequence is not completed, or the working is not guaranteed.

14.18.7 Power-Down mode and ACTIVE mode

Twelve stand-by inputs allow putting independently the different parts of CODEC into Power-Down mode.

When all `SB_*=1` except `SB=0` and `SB_SLEEP=0`, the CODEC is in ACTIVE mode, it's ready for play sound or record sound. But still need follow the anti-pop start or stop sequence. Please refer to "Start up sequence" and "Shutdown sequence".

14.18.8 Working modes summary

Different working modes are sum-up in the following table (non exhaustive table):

Mode	SB	SB_SLEEP	SB_DAC	SB_HP	SB_LO	SB_ADC	SB_MICBIAS1	SB_MICBIAS2	SB_MIC1	SB_MIC2	SB_LBY1	SBLBY2	ADC_IN_SEL	HP_SEL	LO_SEL	MICSTEREO	DAC_LEFT_ONLY	ADC_LEFT_ONLY	DAC_MUTE	HP_MUTE	LO_MUTE
Reset mode (complete power-down mode)	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1
Complete power-down mode	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Sleep mode	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Mono Record Mode, Line input to ADC	0	0	X	X	X	0	X	X	0	10	X	X	00 01	X	X	0	X	1	X	X	X
Stereo Record Mode, Line input to ADC	0	0	X	X	X	0	X	X	0	0	X	X	00	X	X	1	X	0	X	X	X
Stereo Playback mode, DAC to HP	0	0	0	0	X	X	X	X	X	X	X	X	X	000	X	X	0	X	0	0	X
Stereo Playback mode, DAC to LO	0	0	0	X	0	X	X	X	X	X	X	X	X	X	010	X	0	X	0	X	0
Bypass mode, stereo Line to HP	0	0	X	0	X	X	X	X	X	X	0	0	X	010	XX	X	X	X	X	0	X

14.19 SYS_CLK turn-off and turn-on

The main clock of CODEC is called SYS_CLK, which is generated in CPM module and called MCLK. During the SLEEP mode and the complete power-down mode, the main clock SYS_CLK may be stopped to reduce the power consumption to the leakage currents only. In other modes, the main clock SYS_CLK must not be stopped.

The main clock SYS_CLK must not be stopped until CODEC has reached the complete power-down mode and must be restarted before leaving the power-down mode.

After SYS_CLK restarts, it is required to wait 4 SYS_CLK cycles before reading or writing the registers.

When SYS_CLK is turned off (SB_SLEEP=1 or SB=1), writing on register values are not taken into account, register values are not up to date when read and interrupts not generated until SYS_CLK turns on.

14.20 Requirements on outputs and inputs selection and power-down modes

The following rules must be respected in order not to damage performances and to keep the functionality:

- If SB_MIC1 is set to 1, MICSTEREO must be equal to 0, IN_SEL, HP_SEL and LO_SEL must not be equal to '00'.
- If SB_MIC2 is set to 1, MICSTEREO must be equal to 0, IN_SEL, HP_SEL and LO_SEL must not be equal to '01'.
- If SB_LINE is set to 1, IN_SEL must not be equal to '10'.
- If SB_LIBY is set to 1, HP_SEL and LO_SEL must not be equal to '10'.
- If SB_DAC is set to 1, HP_SEL and LO_SEL must not be equal to '11'.

14.21 Anti-pop operation sequences

The main idea of this section is to describe the sequences to perform to minimize the audible pop to the minimum for the headphone output.

Due to the large number of stand-by combinations and to be the most flexible, the handling of the sequence from one working mode to another is left to the software. So for helping the software designer in this task, some specific sequences are automatically performed by CODEC and an interrupt mechanism (IRQ signal and associated registers) warns the application when these sequences end.

14.21.1 Initialization and configuration

To use the embedded CODEC with AIC, the following AIC registers should be set up before start the CODEC:

```
AICFR.ICDC = 1
AICFR.AUSEL = 1
AICFR.BCKD = 0
AICFR.SYNCD = 0
I2SCR.AMSL = 0
I2SCR.ESCLK = 1
```

14.21.2 Start up sequence (DAC)

This sequence is from Power-on mode to CODEC REPLAY mode.

The output sound is **drived** by DAC.

The intent of the following sequence is to prevent for large audible glitches due to the system start-up with the CODEC.

Before this sequence, setup the AIC properly.

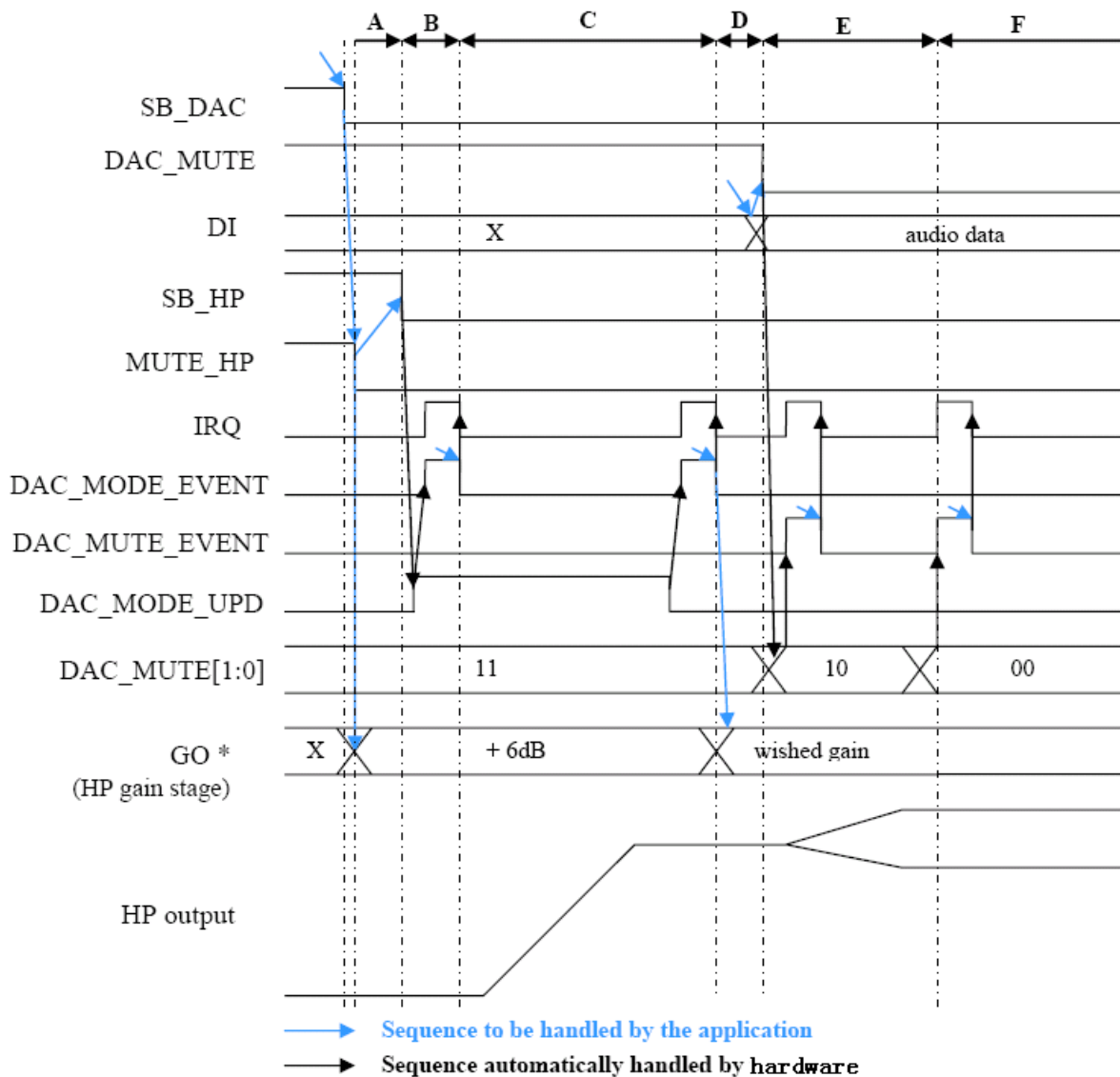


Figure 14-11 Start up sequence

SEQUENCE:

- A : Once the sleep mode is left, the application turns on the DAC (SB_DAC register bit) and after 0.5 ms switches the analog mute signal to '0' (MUTE_HP); then the application sets the HP gain stage to the maximum value (+6 dB)
- B: Ramp-up cycle starts: after waiting 1 ms, the application turns on the headphone output stage (SB_HP = 0). The codec sets DAC_MODE_UPD to '1' and then generates an interrupt on IRQ by setting DAC_MODE_EVENT flag to '1'.
- C: Ramp-up cycle ends: once the ramp-up cycle is started, the application handles the interrupt, resets the DAC_MODE_EVENT flag by writing '1' on it. Once the ramp ends, the Codec sets DAC_MODE_UPD to '0' and then generates an interrupt on IRQ by setting DAC_MODE_EVENT flag to '1'.
- D: IRQ handling and gain-up cycle: the application handles the interrupt, resets the DAC_MODE_EVENT flag by writing '1' on it. The application first sets the wished gain on

HeadPhone and then releases the DAC mute (DAC_MUTE register bit). In the meantime, the application sends valid audio data to the Codec DAC.

E: IRQ generation: once the gain-up cycle starts, DAC_MUTE[1:0] is set to '10' (cf register 4.2.2.8) and the Codec generates an interrupt by setting DAC_MUTE_EVENT flag to '1'. Then the application resets the DAC_MUTE_EVENT flag by writing '1' on it.

F: IRQ handling and active mode: once the gain-up cycle ends, DAC_MUTE[1:0] is set to '00' (cf register 4.2.2.8) and the Codec generates an interrupt by setting DAC_MUTE_EVENT flag to '1'. The application handles the interrupt and resets the DAC_MUTE_EVENT flag by writing '1' on. The Codec DAC path is now fully activated.

14.21.3 Shutdown sequence (DAC)

This sequence is from CODEC REPLAY mode to STANDBY mode.

The output sound is driving by DAC.

The intent of the following sequence is to prevent for large audible glitches due to the system shutdown with the CODEC.

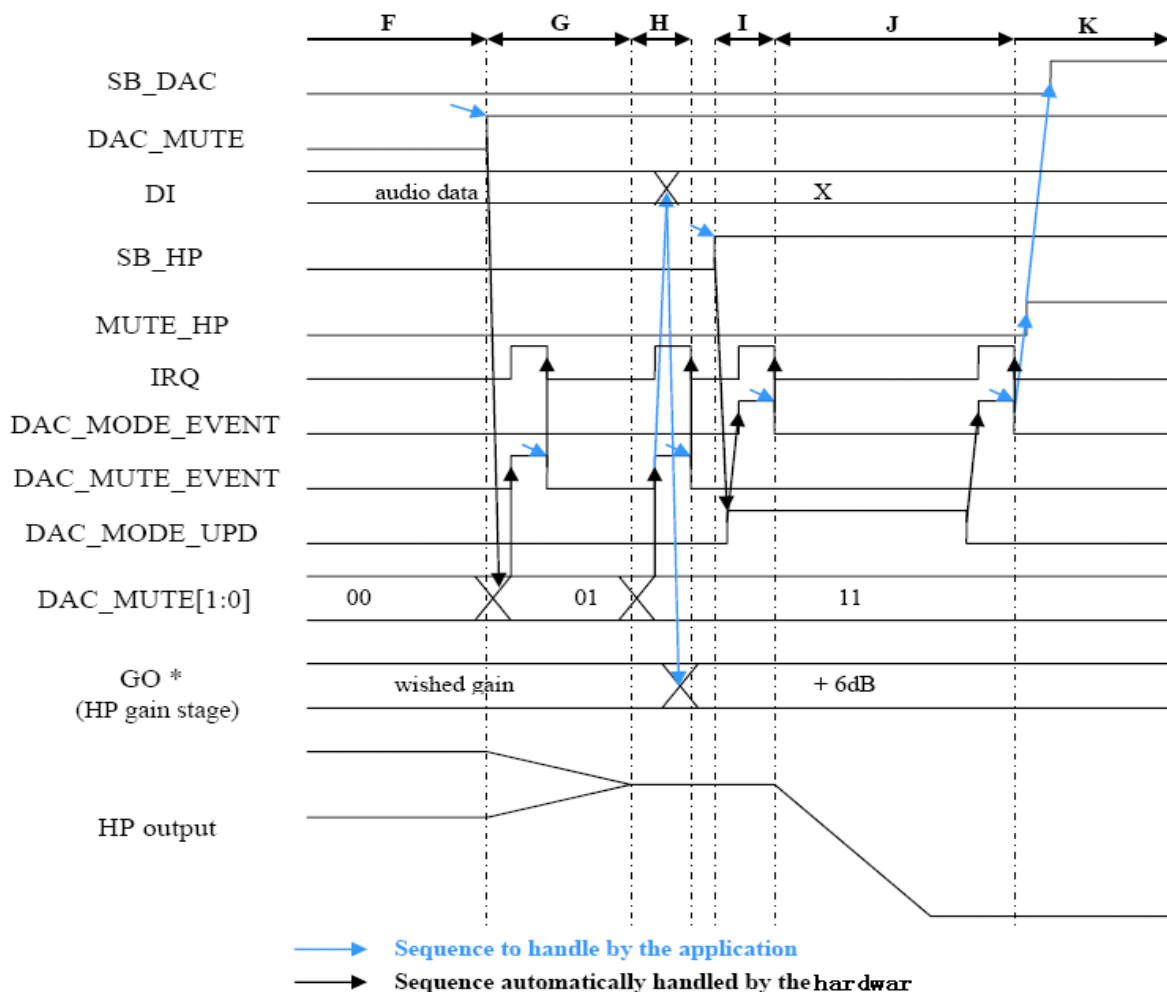


Figure 14-12 Shutdown sequence

SEQUENCE:

F: Initial state: the DAC path is activated.

G: Gain-down cycle starts: the application activates the mute of the DAC (DAC_MUTE register bit).

Once the gain-down cycle starts, DAC_MUTE[1:0] is set to '01' (cf register 4.2.2.8) and the Codec generates an interrupt on IRQ by setting DAC_MUTE_EVENT flag to '1'. Then the application handles the interrupt and resets the DAC_MUTE_EVENT flag by writing '1' on it.

H: Gain-down cycle ends: once the gain-down cycle completes, DAC_MUTE[1:0] is set to '11' (cf register 4.2.2.8) and the Codec generates an interrupt on IRQ by setting DAC_MUTE_EVENT flag to '1'. The application handles the interrupt stopping sending audio data and resets the DAC_MUTE_EVENT flag by writing '1' on it. Then the application sets the HP gain stage to the maximum value (+6 dB).

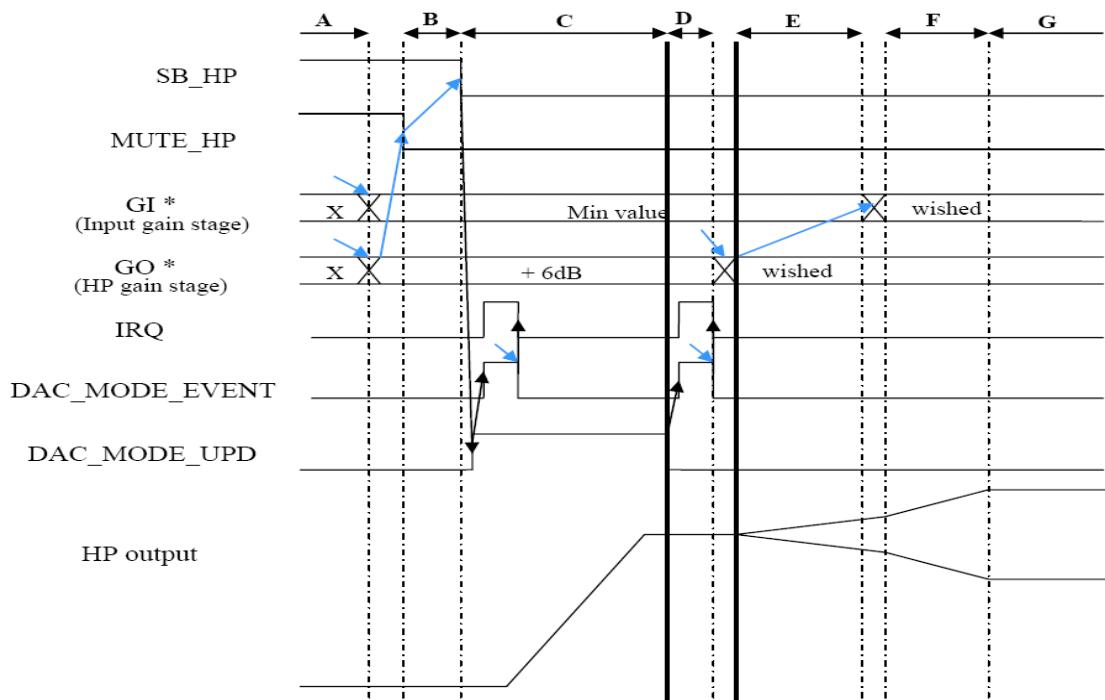
I: Ramp-down cycle starts: the application turns off the headphone output stage (SB_HP = 1). The Codec sets DAC_MODE_UPD to '1' and then generates an interrupt on IRQ by setting DAC_MODE_EVENT flag to '1'. The application handles the interrupt, resets the DAC_MODE_EVENT flag by writing '1' on it.

J: Ramp-down cycle ends: once the ramp down cycle ended, the CODEC sets DAC_MODE_UPD to '0' and then generates an interrupt on IRQ by setting DAC_MODE_EVENT flag to '1'.

K: IRQ handling: the application handles the interrupt and resets the DAC_MODE_EVENT flag by writing '1' on it. Then, the application can activate the analog mute (MUTE_HP). Finally, the application turns off the DAC path (SB_DAC register bit) to be in sleep mode or turn off the Codec (SB_SLEEP, SB register bits).

14.21.4 Start up sequence (Line input)

Headphone port when driven by Line Input

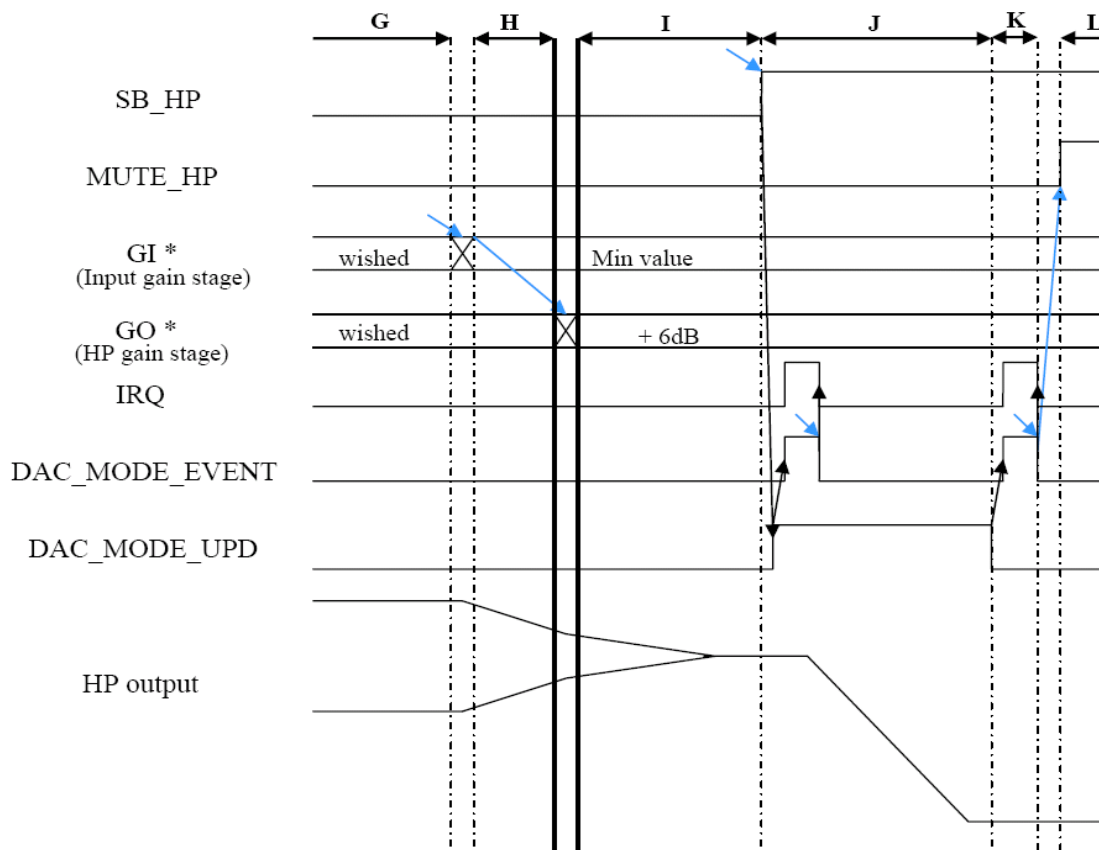


SEQUENCE:

- A: initial state: playback mode is active. Valid analog audio signals are available at the input of the switch matrix
- B: initializing output port: the application first sets the line-in gain to its minimum value and headphone gain stage to its maximum value (+6 dB). This setting is taken into account after a few clocks cycles. Then the application sets MUTE_HP to '0'.
- C: ramp-up cycle: the application turns on the headphone output stage (SB_HP = '0'). The ViC sets DAC_MODE_UPD to '1' and then generates an interrupt on IRQ by setting DAC_MODE_EVENT flag to '1'. Once the ramp-up cycle starts, the application handles the interrupt, resets the DAC_MODE_EVENT flag by writing '1' on it.
- D: IRQ generation: once the ramp-up cycle ends, the ViC sets DAC_MODE_UPD to '0' and generates an interrupt on IRQ by setting DAC_MODE_EVENT flag to '1'. The application resets the DAC_MODE_EVENT flag by writing '1' on it.
- E: Headphone stage gain-up: the application sets HP gain stage to the targeted value (TBD). The maximum duration of the gain ramping equals (number of gain steps * Tcrossout). Refer to section 4.6 for the value of Tcrossout.
- F: Line-in stage gain-up: the application sets the line-in gain stage to the targeted value. The maximum duration of the gain ramping equals (number of gain steps * Tcrossout).
- G: Active mode: the signal path is now fully active.

14.21.5 Shutdown sequence (Line input)

Stopping sequence from playback mode to sleep mode



SEQUENCE:

- G: Active mode: the bypass mode is on.
- H: Line-in stage gain-down: the application sets the line-in gain stage to the minimum value. The maximum duration of the gain ramping equals (number of gain steps * Tcrossout). Refer to section 4.6 for the value of Tcrossout.
- I: Headphone stage gain-up (TBD): the application sets the headphone gain to the maximum value (+6 dB). The maximum duration of the gain ramping equals (number of gain steps * Tcrossout).
- J: Ramp-down cycle: the application can now turn off the headphone output stages (SB_HP register bit). The ViC sets DAC_MODE_UPD to '1' and generates an interrupt on IRQ by setting DAC_MODE_EVENT flag to '1'. The application resets the DAC_MODE_EVENT flag by writing '1' on it.
- K: IRQ generation: once the ramp down cycle ends, the ViC sets DAC_MODE_UPD to '0' and generates an interrupt on IRQ by setting DAC_MODE_EVENT flag to '1'.
- L: IRQ handling: the application handles the interrupt and resets the DAC_MODE_EVENT flag by writing '1' on it. The application can then activate the analog mute (MUTE_HP). The signal path is now off.

14.22 Circuits design suggestions

This section lists a few PCB design suggestions with **different** using mode.

14.22.1 Avoid quiet ground common currents

14.22.1.1 References pins

To work properly, CODEC requires a few additional external components.

CODEC includes an internal voltage reference. To insure a correct common mode biasing of the internal components, an additional voltage VCAP is used. This requires connecting two decoupling capacitors (Cext) between the pin VCAP and AVSCDC. One 10uF low ESR (ceramic or tantalum) and one 100nF ceramic have to be used. The ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch).

14.22.1.2 Power supply pins

CODEC analog power supplies require external decoupling capacitors.

For each power supply pin, one 100nF ceramic capacitor has to be used. This ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch). One low ESR (ceramic or tantalum) capacitor has to be used to decouple the analog power supply provided to the CODEC. Its value depends on the power supply generator; its typical value is between 1uF and 10uF. Ideally use separate ground planes for analog and digital parts.

Connect all ground pins with thick traces to power plane in order to ensure lowest impedance connections.

AVSCDC must be connected to the PCB analog single point reference (star connection) ground (AGND).

14.22.2 Headphone connection (Capacitor-coupled)

Capacitor-coupled headphone and line connection

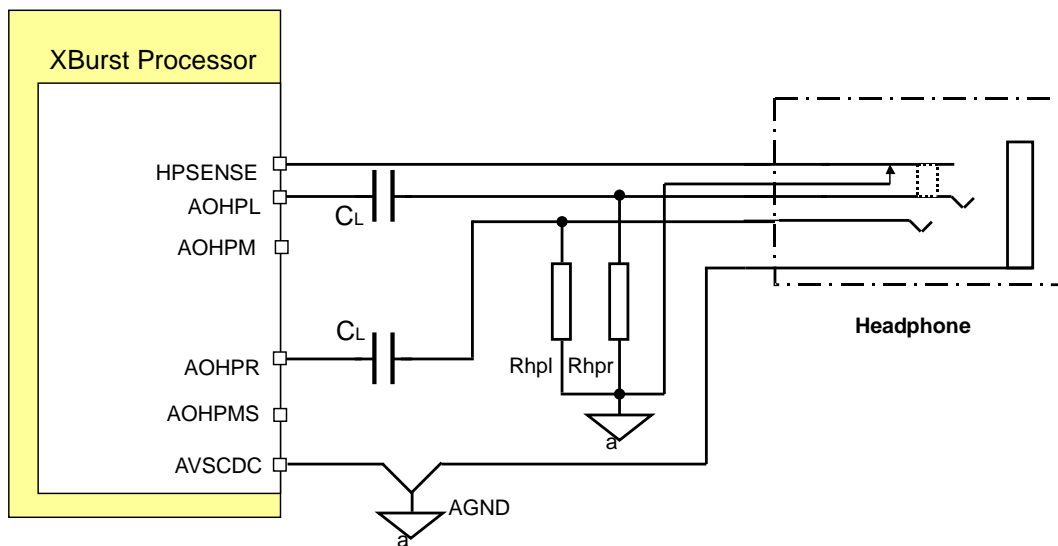


Figure 14-13 Capacitor-coupled connection

The AOHPL and AOHPR pins are connected to the headphone through an external bypass capacitor which is a DC blocking capacitors.

This capacitor is called C_L . When the headphone resistance R_L is 16 Ohm, The tantalum blocking capacitor C_L is 220 μF .

The DC value of the signal AOHPL or AOHPR equals to $AVDCDC/2$.

The ground of the headphone is connected to AGND, which is the PCB analog single point reference (star connection) ground.

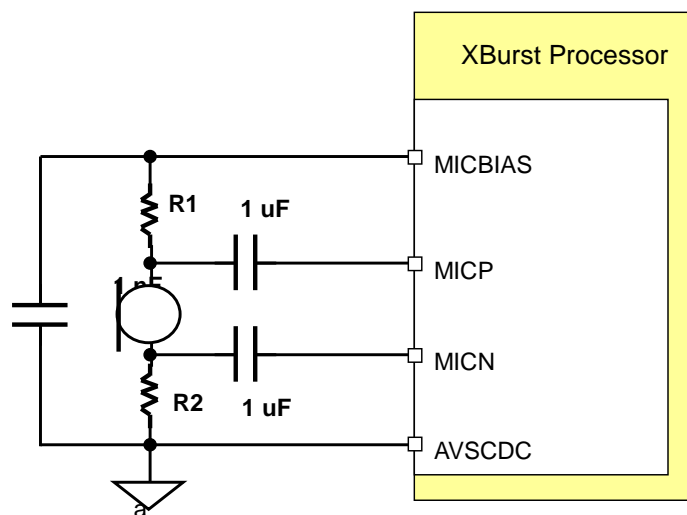
14.22.3 Microphone connection

The optimal performance for the SNR is obtained in differential Microphone inputs with a FS input level

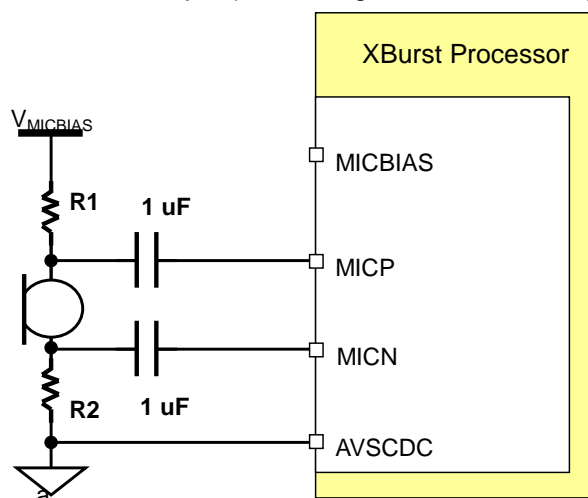
corresponding to peak-to-peak amplitude of the signal is 0.2125V when GIM = 20 (Gain value = 20).

We recommend customer to use differential MIC input for better performance.

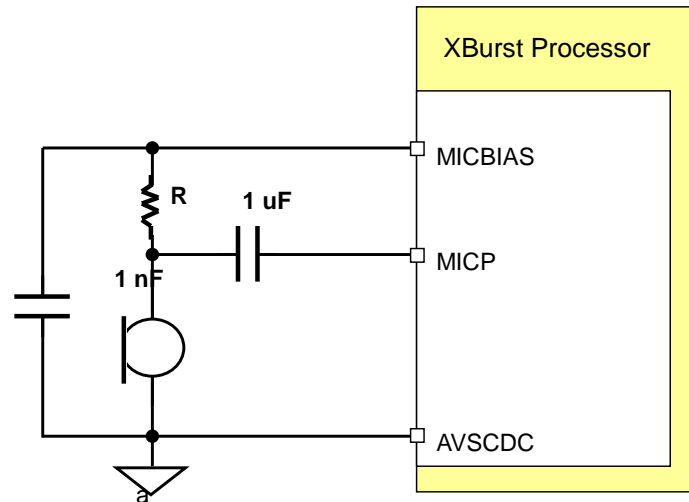
Application schematic with differential MIC input (using MICBIAS pin):



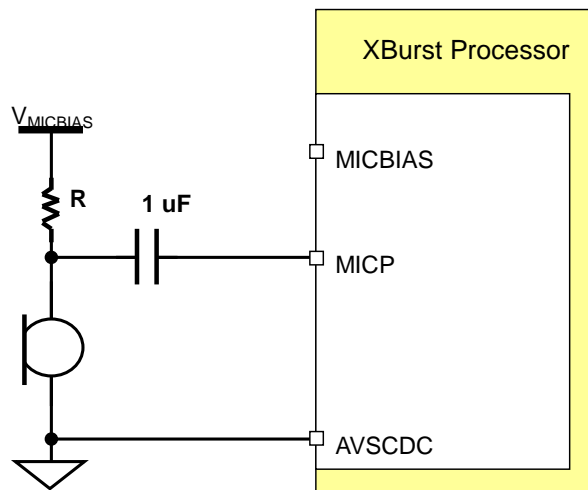
Application schematic with differential MIC input (Vmicbias generated on board):



Application schematic with single-ended MIC input (using MICBIAS pin):



Application schematic with single-ended MIC input ($V_{micbias}$ generated on board):



In single-ended connection, one external resistor (R) has to be used to bias the electret microphone. In differential connection, a pair of external resistor (R_1 , R_2) has to be used to bias the electret microphone. The resistors value relation between them is $R_1 = R_2 = R/2$.

Specific value of resistor (R , commonly from 2.2k Ohm to 4.7k Ohm) and $V_{micbias}$ (if generated on board, usually from 1 to 2V or more) depends on the selected EC (Electret Condenser) microphone. The 1nf decoupling capacitance used in MICBIAS pin removes high frequency noise of the chip.

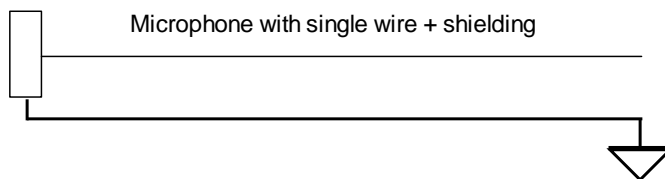
Setting SB_MIC1/SB_MIC2 to 1 will close microphone input path for saving power, also setting $SB_MICBIAS$ to 1 will close MICBIAS stage and the MICBIAS output voltage will be zero.

MICBIAS output voltage scales with $AVDCDC$, equals to $5/6 * AVDCDC$ (typical 2.08V).

MICBIAS output current is 4mA max.

MICBIAS output noise is 40uVrms max.

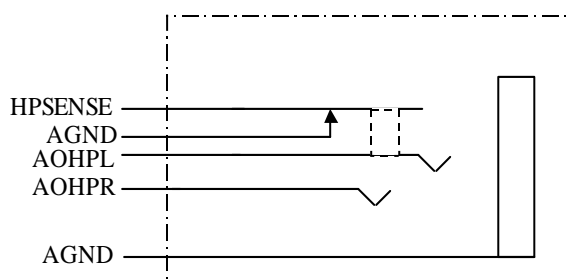
This configuration is better suited for microphone with single wire + shielding.



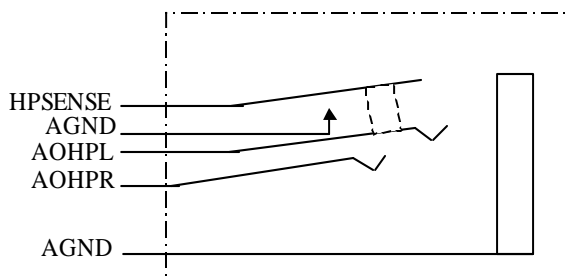
The AVSCDC Pin is connected the analog quiet reference ground in the chip (refers to Grounds and analog signal references). So the ground of MIC must be connected to AVSCDC using a star connection.

14.22.4 Description of the connections to the jack

When the jack is inserted, “sense” and “ground” are disconnected.



No jack plugged: the switch acts as a short-circuit.



Jack plugged: the switch acts as an open circuit.

14.22.4.1 Grounds and analog signal references

In order to limit the parasitic disturbances from the AVSHP output power supplies to inter VREFN analog quiet ground (which is using AVSCDC pin), should use the following principle to distribute the grounds.

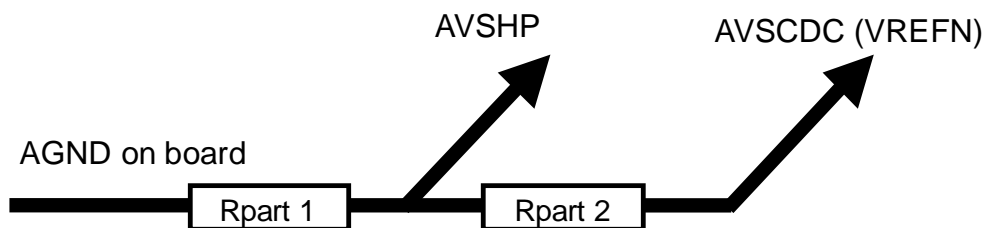


Figure 14-14 Ground distributing

Minimize the values of the connections parasitic resistance Rpar1, Rpar2.

Take a special care for Rpar1 in order to limit the disturbance from the output stages (AVSHP) to the signal reference (VREFN).

The reference of the input signals must be connected to VREFN (internal quiet ground which using the AVSCDC pin) using a star connection.

14.22.5 PCB considerations

To work properly, CODEC analog power supplies require external decoupling capacitors.

In the VCAP pin, one 10uF low ESR (ceramic or tantalum) called C2 and one 100nF ceramic called C1 have to be used. The ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch).

For each power supply pin, one 100nF ceramic capacitor has to be used. The capacitor used in AVDCDC pin is called C4, the capacitor used in AVDHP pin is C6. These ceramic capacitors have to be kept as close as possible to IC package (closer than 0.2 inch).

One low ESR (ceramic or tantalum) capacitor called C3 has to be used to decouple the analog power supply provided to the CODEC. Its value depends on the power supply generator; its typical value is between 1uF and 10uF. Ideally use separate ground planes for analog and digital parts.

C1, C2, C3, C4, C5, C6 are defined in section [“Required external components”](#).

The reference PCB design is shown below:

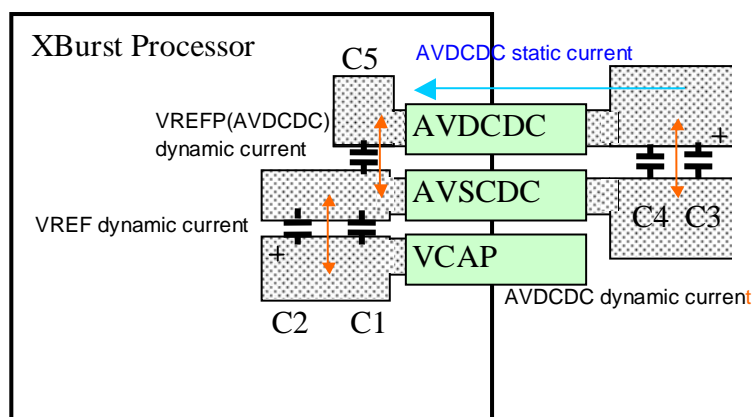


Figure 14-15 the bottom corner of chip PCB Layer

This is just an example reference diagram. You should change and select the PCB layer and route with

your design constraints.

14.22.5.1 Required external components

The following table summarizes the external components required for a proper working of CODEC, except those used for the analog input and output signals.

Name	Description	Typical Value	Unit
C1	Ceramic reference decoupling capacitor. Cext.	100	nF
C2	Tantalum reference decoupling capacitor. Cext.	10	uF
C3	Tantalum analog power supply decoupling capacitor.	1 to 10	uF
C4	Ceramic AVDCDC decoupling capacitor.	100	nF
C5	Ceramic inter signal VREFP decoupling capacitor (can be shared with C4).	100	nF
C6	Ceramic AVDHP decoupling capacitor. Not shown in section PCB considerations .	100	nF
C8	MICBIAS decoupling capacitor, Refer to section "Microphone connection" .	1	nF
C9, C10	External bypass capacitor, for DC blocking, Refer to section "Headphone connection (Capacitor-coupled)" .	220	uF
Rhpl, Rhpr	Headphone jack pull-down resistors, Refer to section "Headphone connection (Capacitor-coupled)" .	470 or 4.7K	Ohm
R	In single-ended connection, one external resistor (R) has to be used to bias the electret microphone. Refer to section "Headphone connection (Capacitor-coupled)" .	2.2K ~ 4.7K	Ohm

14.23 Analog characteristics

14.23.1 Line input to audio ADC path

Measurement conditions:

T = 25°C, AVDCDC = 3.3 V, input sine wave with a frequency of 1 kHz, Fmclk = 12 MHz, Fs = 8 to 96 kHz, measurement bandwidth 20 Hz – 20 kHz, unless otherwise specified.

Parameter	Test conditions	Min.	Typ	Max.	Unit
Input level (2)	Full Scale, Gain GIDL, GIDR = 0 dB, boost gain GIM1, GIM2 = 20 dB	0.189	0.212	0.239	Vpp
SNR	A-weighted, 1 kHz sine wave @ Full Scale and gain GIDL, GIDR = 0 dB, boost gain GIM1, GIM2 = 0 dB	85 (TBC)	90 (TBC)		dB

	A-weighted, 1 kHz sine wave @ Full Scale and gain GIDL, GIDR = 0 dB, boost gain GIM1, GIM2 = 20 dB	75 (TBC)	80 (TBC)		dB
THD	1 kHz sine wave @ Full Scale -1 dB and gain GIDL, GIDR = 0 dB, boost gain GIM1, GIM2 = 20 dB		-70 (TBC)	-60 (TBC)	dB
	1 kHz sine wave @ Full Scale -1 dB and gain GIDL, GIDR = 0 dB, boost gain GIM1, GIM2 = 0 dB		-80 (TBC)	-70 (TBC)	dB
Dynamic range	A-weighted, 1 kHz sine wave @ Full Scale -60 dB and gain GIDL, GIDR = 0 dB, boost gain GIM1, GIM2 = 0 dB (note 1)	85 (TBC)	90 (TBC)		dB
	A-weighted, 1 kHz sine wave @ Full Scale -60 dB and gain GIDL, GIDR = 0 dB, boost gain GIM1, GIM2 = 20 dB (note 1)	75 (TBC)	80 (TBC)		dB
PSRR	100 mVpp 1 kHz sinewave is applied to AVDCDC, input data is 0 and gain GIDL, GIDR = 0 dB, boost gain GIM1, GIM2 = 20 dB		90 (TBC)		dB
Gain boost	Boost gain GIM1, GIM2 when activated	0		20	dB
Gain boost step	GIM1, GIM2 @1 kHz		4		dB
Gain boost accuracy	GIM1, GIM2 @1 kHz	-1 (TBC)		+1 (TBC)	dB
Input resistance	Boost gain GIM1, GIM2 = 0 dB	63	80	96	kOhm
	Boost gain GIM1, GIM2 = 20 dB	10	12.5	15	
Input capacitance	Includes 10 pF for ESD, bonding and package pins capacitances			25	pF
Input bypass capacitor	Cbyline		1		uF

Note 1: The specified value is extrapolated by adding 60 dB to the measured SNR

Note 2: The Full Scale input voltage scales with LDO output: VREFP_CODEC.

14.23.2 Audio DAC to headphone output path

Measurement conditions:

T = 25°C, AVDCDC = 3.3 V, input sine wave with a frequency of 1 kHz, Fmclk = 12 MHz, Fs = 8 to 96 kHz, measurement bandwidth 20 Hz – 20 kHz, unless otherwise specified.

Parameter	Test conditions	Min.	Typ	Max.	Unit
FOutput level (3)	ull Scale, Gain GOL, GOR = 0 dB, GODL, GODR = 0 dB, 10 kOhm load	1.89	2.12	2.39	Vpp

	Full Scale, Gain GOL, GOR = -3 dB, GODL, GODR = 0 dB, 16 Ohm load (TBC)	1.33	1.5	1.69	Vpp
Maximum output power	RI = 16 Ohm		17.6 (TBC)		mW
SNR	A-weighted, 1 kHz sine wave @ Full Scale, Gain GOL, GOR = 0 dB, GODL, GODR = 0 dB, 10 kOhm load	95 (TBC)	100 (TBC)		dB
Idle Noise	A-weighted with no signal and gain GOL, GOR = -10 dB, GODL, GODR = 0 dB, 16 Ohm load		-103 (TBC)	-98 (TBC)	dBV
THD, THD+N	1 kHz sine wave @ Full Scale -1 dB, GOL, GOR = 0 dB, GODL, GODR = 0 dB, 10 kOhm load		-85 (TBC)	-75 (TBC)	dB
	1 kHz sine wave @ Full Scale -1 dB and gain GOL, GOR = -3 dB, GODL, GODR = 0 dB, 16 Ohm load		-70 (TBC)	-65 (TBC)	dB
Dynamic range	A-weighted, 1 kHz sine wave @ Full Scale -60 dB, Gain GOL, GOR = 0 dB, GODL, GODR = 0 dB, 10 kOhm load (note 1)	95 (TBC)	100 (TBC)		dB
Wide Band Noise	1 kHz sine wave @ Full Scale and gain GOL, GOR = 0 dB, GODL, GODR = 0 dB, 10 kOhm load, measurement bandwidth 20 kHz – 100 kHz		65		dB
PSRR	100 mVpp 1 kHz sinewave is applied to AVDCDC, input data is 0 and gain GOL, GOR = 0 dB, GODL, GODR = 0 dB, 10 kOhm load		90 (TBC)		dB
	100 mVpp 1 kHz sinewave is applied to AVDCDC, input data is 0 and gain GOL, GOR = -25 dB, GODL, GODR = 0 dB, 16 Ohm load		90 (TBC)		dB
PuN	Active <-> inactive, 10 kOhm load		-60 (TBC)		dBVp
	Active <-> inactive, 16 Ohm load		-60 (TBC)		dBVp
Output resistance	RI	16			Ohm
Output bypass capacitor	CI (RI = 10 kOhm)			1	uF
	CI (RI = 16 Ohm)			220	uF
Gain range	GOL, GOR 5-bit programmable range, digital control, @1 kHz	-25		+6	dB

	GODL, GODR 5-bit programmable range, digital control, @1 kHz	-31		0	
Gain step	GOL, GOR and GODL, GODR monotonic		1		dB
Gain accuracy	GOL, GOR, GODL, GODR @1 kHz	-0.5		+0.5	dB
Output capacitance (note 2)	Cp			200	pF
Polarity	DIL,R to AOHPL,R		+1		

Note 1: The specified value is extrapolated by adding 60 dB to the measured SNR with a FS-60 dB input signal.

Note 2: Output may oscillate above specified load capacitances. The capacitance is equivalent to a 2-meter cable.

Note 3: The Full Scale input voltage scales with the nLR output: VREFP_CODEC.

14.23.3 Audio DAC to mono line output path

Measurement conditions: T = 25°C, AVDCDC = 3.3 V, input sine wave with a frequency of 1 kHz, Fmclk = 12 MHz, Fs = 8 to 96kHz, measurement bandwidth 20 Hz – 20 kHz, unless otherwise specified.					
Parameter	Test conditions	Min.	Typ	Max.	Unit
Output level (2)	Full Scale, Gain GODL, GODR = 0 dB	3.78	4.25	4.78	Vpp
SNR	A-weighted, 1 kHz sine wave @ Full Scale, Gain GODL, GODR = 0 dB	90 (TBC)	95 (TBC)		dB
THD+N	1 kHz sine wave @ Full Scale -1 dB, Gain GODL, GODR = 0 dB		-85 (TBC)	-75 (TBC)	dB
Dynamic range	A-weighted, 1 kHz sine wave @ Full Scale – 60 dB, Gain GODL, GODR = 0 dB (note 1)	90 (TBC)	95 (TBC)		dB
PSRR	100 mVpp 1 kHz sinewave is applied to AVDCDC, Gain GODL, GODR = 0 dB, input data is 0		90 (TBC)		dB
Output resistance	RI	10			kOhm
Output capacitance	Cp			100	pF

Note 1: The specified value is extrapolated by adding 60 dB to the measured SNR

Note 2: The Full Scale input voltage scales with the nLR output: VREFP_CODEC.

14.23.4 Line input to headphone output path (analog bypass)

Measurement conditions: T = 25°C, AVDCDC = 3.3 V, input sine wave with a frequency of 1kHz, Fmclk = 12 MHz, Fs = 8 to 96 kHz, measurement bandwidth 20 Hz – 20 kHz, unless otherwise specified.					
Parameter	Test conditions	Min.	Typ	Max.	Unit
Input level	Full Scale	1.89	2.12	2.39	Vpp
Input resistance		10k			Ohm
Output level (2)	Full Scale, Gain GOL, GOR = 0 dB, GIL, GIR = 0 dB, 10 kOhm load (note 1)	1.89	2.12	2.39	Vpp
	Full Scale, Gain GOL, GOR = -3 dB, GIL, GIR = 0 dB, 16 Ohm load (TBC)	1.33	1.5	1.69	Vpp
SNR	A-weighted, 1 kHz sine wave @ Full Scale, Gain GOL, GOR = 0 dB, GIL, GIR = 0 dB	90 (TBC)	95 (TBC)		dB
THD, THD+N	1 kHz sine wave @ Full Scale -1 dB, Gain GOL, GOR = 0 dB, GIL, GIR = 0 dB, 10 kOhm load		-85 (TBC)	-75 (TBC)	dB
	1 kHz sine wave @ Full Scale -1 dB and gain GOL, GOR = -3 dB, GIL, GIR = 0 dB, 16 Ohm load		-70 (TBC)	-65 (TBC)	dB
Dynamic range	A-weighted, 1 kHz sine wave @ Full Scale -60 dB, Gain GOL, GOR = 0 dB, GIL, GIR = 0 dB, 10 kOhm load (note 1)	90 (TBC)	95 (TBC)		dB
PSRR	100 mVpp 1 kHz sinewave is applied to AVDCDC, input data is 0 and gain GOL, GOR = 0 dB, GIL, GIR = 0 dB, 10 kOhm load		90 (TBC)		dB
	100 mVpp 1 kHz sinewave is applied to AVDCDC, input data is 0 and gain GOL, GOR = -25 dB, GIL, GIR = 0 dB, 16 Ohm load		90 (TBC)		dB
PuN	Active <-> inactive, 10 kOhm load		-60 (TBC)		dBVp
	Active <-> inactive, 16 Ohm load		-60 (TBC)		dBVp
Output resistance	RI	16			Ohm
Gain range	GIL, GIR 5-bit programmable range, digital control, @1 kHz	-25		+6	dB
Gain step	GIL, GIR monotonic		1		dB
Gain accuracy	GIL, GIR, @1 kHz	-0.5		+0.5	dB
Input capacitance	Includes 10 pF for ESD, bonding and package pins capacitances			25	pF

Input bypass capacitor	Cbyline		1		uF
Polarity	ALL,R to AOL,R		+1		

Note 1: The specified value is extrapolated by adding 60 dB to the measured SNR

Note 2: The Full Scale input voltage scales with the nLR output: VREFP_CODEC

14.23.5 Micbias and reference

Measurement conditions:
 T = 25°C, AVDCDC=3.3 V, input sine wave with a frequency of 1 kHz, Fmclk = 12 MHz, Fs = 8 to 96 kHz, measurement bandwidth 20 Hz – 20 kHz, unless otherwise specified.

Parameter	Test conditions	Min.	Typ	Max.	Unit
Micbias output level		2.35 (TBC)	2.5 (TBC)	2.65 (TBC)	V
Micbias output current				4	mA
Micbias output noise	A-weighted		30 (TBC)	40 (TBC)	uVrms
Micbias decoupling capacitor	Cmic	0.75	1	1.25	nF
VCAP voltage			2 (TBC)		V
VREFP_CODEC (also called VREF)		2.35 (TBC)	2.5 (TBC)	2.65 (TBC)	V

Section 4

MEMORY INTERFACE

15 DDR Controller

15.1 Overview

DDRC (DDR Controller) is a general IP which provides an interface to DDR2, DDR3, mobile DDR(LPDDR), LPDDR2 memory. The DDRC IP is designed for SOC usage and is configurable, scalable to meet the requirement of various SOC.

Features:

- Multi-port Architecture and asynchronous interface to all port
- Support clock-stop mode
- Support auto-refresh and self-refresh
- Support power-down mode and deep-power-down mode
- Programmable DDR timing parameters
- Programmable DDR row and column address width
- Programmable remapping from Logic address to Physical address

15.1.1 Supported DDR SDRAM Types

- DDR2
- DDR3
- mobile DDR(LPDDR)
- LPDDR2

Row address width less than 16-bit & Column width less than 12-bit are supported.

15.1.2 Block Diagram

Following figure shows the functional block diagram of DDRC.

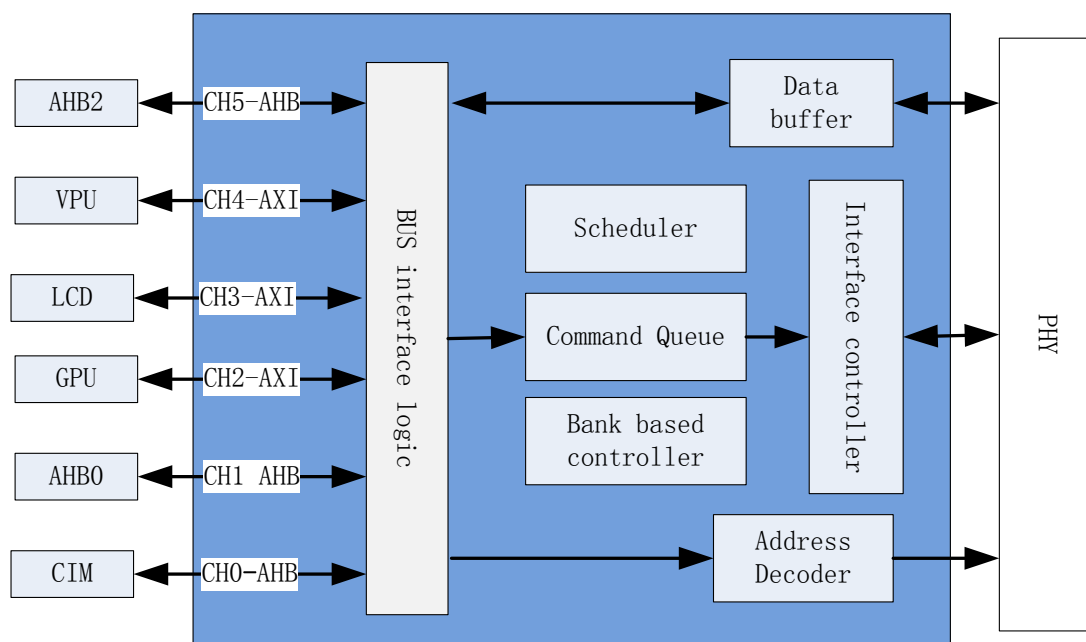


Figure 15-1 DDRC block diagram

15.2 Register Description

Following table lists the registers of DDR Controller. All of these registers are 32bit

All DDRC register can be accessed by channel 1: CH1_AHB.

The physical address base is 0x13010000.

Table 15-1 DDRC Register

Name	Address offset	Width	Access	Description
DSTATUS	0x00	32	RW	DDR Status Register
DCFG	0x04	32	RW	DDR Configure Register
DCTRL	0x08	32	RW	DDR Control Register
DLMR	0x0C	32	RW	DDR Load-Mode-Register
DTIMING1	0x60	32	RW	DDR Timing Configure Register 1
DTIMING2	0x64	32	RW	DDR Timing Configure Register 2
DTIMING3	0x68	32	RW	DDR Timing Configure Register 3

DTIMING4	0x6C	32	RW	DDR Timing Configure Register 4
DTIMING5	0x70	32	RW	DDR Timing Configure Register 5
DTIMING6	0x74	32	RW	DDR Timing Configure Register 6
DREFCNT	0x18	32	RW	Auto-Refresh Counter
DMMAP0	0x24	32	RW	DDR Memory CS0 Map Configure Register
DMMAP1	0x28	32	RW	DDR Memory CS1 Map Configure Register
DDLDP	0xBC	32	RW	DFI low power handshake register
DREMAP1	0x9C	32	RW	DDR address remapping register1
DREMAP2	0xA0	32	RW	DDR address remapping register2
DREMAP3	0xA4	32	RW	DDR address remapping register3
DREMAP4	0xA8	32	RW	DDR address remapping register4
DREMAP5	0xB0	32	RW	DDR address remapping register5
DSTRB	0x34	32	RW	Multi-media stride register
WCMDCTRL1	0x100	32	RW	Write command reorder & grouping (Performance control)
RCMDCTRL0	0x104	32	RW	Read Channel mode control (Performance control)
RCMDCTRL1	0x108	32	RW	Read Channel mode control (Performance control)
WDATTHD0	0x114	32	RW	Wdata Channel mode control (Performance control)
WDATTHD1	0x118	32	RW	Wdata Channel mode control (Performance control)
IPORTPRI	0x128	32	RW	Configuration of Internal Priority (Performance control)
CHDOS	0x07c 0x080 0x084 0x088 0x08c 0x090	32	RW	QoS configure for each channel (Performance control)

	0x094			
CPM_DRCG	0xB0000 0D0	32	RW	CPM register, DLL & Clock control for DDR

15.2.1 DSTATUS

DSTATUS																0x13010000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																ENDIAN	MISS	DPDN	PDN	AREF	SREF	Reserved	CKE0								
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31~8,1: Reserved. Write has no effect, read value not need to care.

ENDIAN: Read-only, indicate the data endian status.

Bit [7]	Description	Remark
0	Little data Endian.	(reset value)
1	Big data Endian.	

MISS: Indicate the accessed address is out of memory mapping range. (This bit is writable)

Bit [6]	Description	Remark
0	No operation miss DDRC memory mapping.	(reset value)
1	At last one operation miss DDRC memory mapping.	

DPDN: Indicate the deep-power-down status of DDR memory.

Bit [5]	Description	Remark
0	DDR memory is NOT in deep-power-down state.	(reset value)
1	DDR memory is in deep-power-down state.	

PDN: Indicate the power-down status of DDR memory.

Bit [4]	Description	Remark
0	DDR memory is NOT in power-down state.	(reset value)
1	DDR memory is in power-down state.	

AREF: Indicate the auto-refresh status of DDR memory.

Bit [3]	Description	Remark
0	DDR memory is NOT in auto-refresh state.	(reset value)
1	DDR memory is in auto-refresh state.	

SREF: Indicate the self-refresh status of DDR memory.

Bit [2]	Description	Remark
0	DDR memory is NOT in self-refresh state.	(reset value)
1	DDR memory is in self-refresh state.	

CKE1: not support in this version.

Bit [1]	Description	Remark
0	CKE1 Pin is low.	(reset value)
1	CKE1 Pin is high.	

CKE0: Indicate the CKE0 Pin status of DDR memory.

Bit [0]	Description	Remark
0	CKE0 Pin is low.	(reset value)
1	CKE0 Pin is high.	

15.2.2 DCFG

Configure the external memory, static configuration only. ie. This register can NOT be changed on-the-fly.

DCFG																0x13010004																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Reserved			ROW1			COL1		BA1	IMBA	BSL	Reserved		TYPE	ODTEN	MISPE	Reserved		COL1			COL0			CS1EN	CS0EN				CL		BA0	DW
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31, 30, 20, 14: Reserved. Writing has no effect, read as zero.

MISPE: Missing CS protection Enable . Set 1 to enable.

If software read (or write) a memory space which is not selected by any CS, controller will return random data to a read operation (or mask write operation) to avoid system bus being locked. A CS missing flag will set in DSTATUS.

BSL: Burst length for DDR chips

1: 8 burst

0: 4 burst

IMBA:

0: CS0, CS1 connected 2 memory chips which have same ROW, COL, BA configuration.

In this mode, both chips are configured by ROW0, COL0, BA0 .

ROW1, COL1, BA1 does not effect the system.

1: CS0, CS1 connected 2 memory chips which have different ROW, COL, BA configuration.

Chip0 is configured by ROW0, COL0, BA0;

Chip1 is configured by ROW1, COL1, BA1;

TYPE: Select external memory device type.

Bit [19:17]	Description	Remark
000	Normal SDR (Single-Data-Rate) SDRAM(Not support).	(reset value)
001	Mobile SDR(Not support).	
010	Normal DDR1 (Not support)	
011	Mobile DDR(LPDDR).	
100	Normal DDR2.	
101	LPDDR2	
110	Normal DDR3	
111	LPDDR3 (Not support).	

ROW0/1: Row Address width. Specify the row address width of external DDR.

	Description	Remark
000	12-bit row address	(reset value)
001	13-bit row address	
010	14-bit row address	
011	15-bit row address	
100	16-bit row address	
Reserved		

COL0/1: Column Address width. Specify the Column address width of external DDR.

	Description	Remark
000	8-bit Column address.	(reset value)
001	9-bit Column address	
010	10-bit Column address	

011	11-bit Column address	
100	12-bit Column address	
Reserved		

CS1EN: DDR Chip-Select-1 Enable.

If an DDR memory chip is connected to ddr pin cs1, set CS1EN=1.

Bit [7]	Description	Remark
0	DDR Pin CS1 is not in use.	(reset value)
1	DDR Pin CS1 is in use.	

CS0EN: DDR Chip-Select-0 Enable.

If DDR memory is connected to ddr pin cs0, set CS0EN=1.

Bit [6]	Description	Remark
0	DDR Pin CS0 is not in use.	(reset value)
1	DDR Pin CS0 is in use.	

CL: CAS Latency.

Bit [5:2]	Description	Remark
Reserved	Not in use for this version	

BA0/1: Bank Address width of DDR memory.

Bit [1]	Description	Remark
0	4 bank device, Pin ba[1:0] is valid, ba[2] un-used.	(reset value)
1	8 bank device, Pin ba[2:0] is valid.	

DW: External DDR Memory Data Width.

Specify the external DDR memory data width.

Bit [0]	Description	Remark
0	External memory data width is 16-bit.	(reset value)
1	External memory data width is 32-bit.	

15.2.3 DCTRL

On the positive edge of START, the command selected by CMD field will be performed.

DCTRL																0x13010008																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								DFI_RST	DLL_RST	CTL_RST	CFG_RST	Reserved	KEEPSR	Reserved	ACTPD	PDT			ACTSTP	Reserved				DPD	SR	UNALIGN	ALH	Reserved	CKE	RESET	
RST	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit 31~24, 19,18,16,10~7,2: Reserved. Writing has no effect, read as zero.

CFG_RST: Reserved, please keep the default value.

CTL_RST: Reserved, please keep the default value.

DLL_RST: Reserved, please keep the default value.

DFI_RST: Reserved, please keep the default value.

KEEPSR: inner usage, keep to 0.

ACTSTP: Active Clock-Stop.

0: Clock can be stopped only after all banks be precharged

1: Clock can be stopped after some bank’s row activated

ACTPD: Active Power-Down.

Some SDRAM devices support Active-Power-Down.

By default, ACTPD=0, hardware will precharge all active banks before entering Power-Down mode, so called Precharge-Power-Down.

By setting ACTPD=1, hardware drives SDRAM into Power-Down mode without precharge all active banks, some banks are still active in Power-Down mode, so called Active-Power-Down.

Bit [15]	Description	Remark
0	Precharge all banks before entering power-down.	(reset value)
1	Do not precharge all banks before entering power-down.	

PDT: Power-Down Timer.

When there’s no access to DDR memory for a period of time, hardware drives DDR into power-down mode to save power consumption. Hardware can exit Power-Down mode automatically when new access arrives.

If PDT=0, power-down function is disabled.

If enable power-down mode, it is recommended to be set after DDR initialization.

Bit [14:13]	Description	Remark
000	power-down disabled, hardware never drive SDRAM into power-down mode.	(reset value)
001	Enter power-down after 8 tCK idle.	
010	Enter power-down after 16 tCK idle.	

011	Enter power-down after 32 tCK idle.	
100	Enter power-down after 64 tCK idle.	
101	Enter power-down after 128 tCK idle.	
110 - 111	Reserved.	

SR: Software drive external DDR device entering Self-Refresh mode.

Software set SR=1 drive external DDR device entering self-refresh mode;

Software set SR=0 drive external DDR device exiting self-refresh mode;

In this mode, the CK to external DDR device would be stopped during self-refresh period;

But the clock supplied to ddr_controller logic would not stop.

Software can read & write ddr_controller registers in this mode.

Software can NOT read or write memory data in this mode.

NOTE: Software must guarantee that there's no memory access during self-refresh mode. Otherwise, software can NOT exit this mode,

system would be hang-up!!

Bit [5]	Description	Remark
0	Drive external DDR device entering self-refresh mode.	(reset value)
1	Drive external DDR device exiting self-refresh mode.	

DPD: Software drives external Mobile DDR device entering Deep-Power-Down mode.

Software set DPD = 1 drives external Mobile DDR device entering Deep-Power-Down mode instead of Power-Down mode, when there's no access to DDR memory for a period of time. So you must firstly enable Power-Down mode (refer to PDT).

Software needs to reset DDR controller and re-do a complete initial process to exit Deep-Power-Down mode.

When external device goes to Deep-Power-Down mode, it lose all data store in memory and registers.

The memory chip will cut off inner power supply for power saving.

UNALIGN: Enable unaligned transfer on AXI BUS.

Bit [4]	Description	Remark
Reserved	Not use in this version	

ALH: Advanced Latency Hiding.

This is a test-oriented register.

Some latency is reduced in special cases.

Bit [3]	Description	Remark
0	Disable ALH.	(reset value)
1	Enable ALH.	

CKE: Control the status of CKE pin.

Write CKE=1 can set CKE pin to HIGH state.

Write CKE=0 would be ignored.

The default value of CKE Pin is low;

CKE0,1 Pins status is represented by DDR_STATUS register.

Caution: This register is used only for DDR initializing sequence; software should NOT update this register when DDR memory is in normal working mode.

Bit [1]	Description	Remark
0	Not set CKE Pin High.	(reset value)
1	Set CKE Pin to HIGH.	

RESET: Module reset for ddr_controller.

Software is able to reset ddr_controller by setting RESET bit high and ends the reset by clear this bit.

Bit [0]	Description	Remark
0	End resetting ddr_controller.	(reset value)
1	Resetting ddr_controller.	

15.2.4 DLMR

DLMR register is used for initializing the DDR SDRAM memory device.

On the positive edge of START, the command selected by CMD field will be performed.

DLMR																0x1301000C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DDR_ADDR														Reserved	BA		Reserved	CMD		Reserved	START										
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 7~6, 2~1: Reserved. Writing has no effect, read as zero.

DDR_ADDR: When performing a DDR command, DDR_ADDR[13:0] corresponding to external DDR address Pin A[13:0]; DDR_ADDR[15:14] are reserved.

Bit [31:12]	Description	Remark
0000_0000	{MA[9:0],OP[9:0]} for LPDDR2 For other DDR the low bit 14 bit corresponding to external DDR address Pin A[13:0].	(reset value)

BA: Bank Address.

When performing a DDR command, BA[2:0] corresponding to external DDR address Pin BA[2:0].

Bit [10:8]	Description	Remark
000	corresponding to external DDR address Pin BA[2:0].	(reset value)

CMD: Select command to process when setting START from low to high.

On the positive edge of START, one of the following commands will be performed.

Bit [5:3]	Description	Remark
000	Precharge one bank / All banks. (dependent field : BA, DDR_ADDR)	(reset value)
001	Auto-Refresh.	
010	Load Mode Register. (dependent field : BA, DDR_ADDR)	
010	ZQCS for DDR3	
100	ZQCL for DDR3	
101~111	Reserved.	

START: Start to perform a command to external DDR memory.

The command is performed on the positive edge of START; Hardware will clear START bit to zero when command issued out to external DDR memory.

Write 0 to START will be ignored and take no effect;

START=1 means hardware is busy executing current command and can NOT accept new command;

Software must check START=0 before writing 1 to START.

Bit [0]	Description	Remark
0	No command is performed.	(reset value)
1	On the positive edge of START, perform a command defined by CMD field.	

15.2.5 DTIMING1,2,3,4,5,6 (DDR Timing Configure Register)

The timing parameters are identical to the JEDEC DDR Specification.

DTIMMING1																0x13010060																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved		tRTP						Reserved		tWTR						Reserved		tWR						Reserved		tWL					
RST	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

DTIMMING2																0x13010064																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved		tCCD						Reserved		tRAS						Reserved		tRCD						Reserved		tRL					
RST	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

DTIMMING3																0x13010068																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved		ONUM				tCKSRE		Reserved		tRP						Reserved		tRRD						Reserved		tRC					
RST	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1

DTIMING4																0x1301006C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved		tRFC						tEXTRW		tRWCOV		tCKE		Reserved				tMINSR				Reserved		tXP		Reserved		tMRD			
RST	0	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	1

DTIMMING5																0x13010070																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	tCTLUPD								Reserved		tRTW						Reserved		tRDLAT						Reserved		tWDLAT					
RST	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

DTIMMING6																0x13010074																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

	tXSRD								Reserved		tFAW				Reserved		tCFGW				Reserved		tCFGR										
RST	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

If the DDR clock period are tCK.

All these registers's value equal to the number tCK cycle. For example, if tRTP=2, it means 2 tCK between READ and PRECHARGE;

tRTP: READ to PRECHARGE command period.

tWTR: WRITE to READ command delay.

tWR: WRITE Recovery Time defined by register MR of DDR2 DDR3 memory.

tWL: Write latency, please notice that this version only support AL=0.

tCCD: CAS# to CAS# command delay

tRAS: ACTIVE to PRECHARGE command period.

tRAS defines the ACTIVE to PRECHARGE command period to the same bank.

tRCD: ACTIVE to READ or WRITE command period.

tRCD defines the ACTIVE to READ/WRITE command period to the same bank.

tRL: Read latency, please notice that this version only support AL=0.

ONUM: Keep to 4 in this version.

tRP: PRECHARGE command period

tRP defines the PRECHARGE to next command period to the same bank.

tRRD: ACTIVE bank A to ACTIVE bank B command period.

tRRD defines the ACTIVE to ACTIVE command period to **different** banks.

tRC: ACTIVE to ACTIVE command period.

tRC defines the ACTIVE to ACTIVE command period to the same bank.

tWDLAT: tWL-1 (when use LPDDR2, set tWDLAT=tWL).

tRDLAT: tRL-2. (when use LPDDR2, set tRDLAT=tRL).

tFAW: 4-active command window.

tCFGW: Write PHY configure registers to other commands delay. Not need to change.

tCFGR: Ready PHY configure registers to other commands delay. Not need to change.

tRTW: Read to Write latency.

tCTLUPD: Inner usage. Not need to change.

tEXTRW: keep the default value.

tRWCOV: keep the default value.

All these registers' value has different rate between the tCK cycle

tXSRD: exit self-refresh to READ delay.

Delay time is tXSRD*4 (tCK)

Note: You can set this registers as tXS(or tXSNR), but must make sure the system do not read ddr before tXSRD in wakeup.

tCKSRE: Valid clock after enter self-refresh($tCKSRX = tCKSRE$ in this version). Delay time is $tCKSRE * 8$ (tCK)

tCKE: minimum CKE pulse width.

tCKE define the minimum CKE pulse width, include high level and low level.

Bit [18:16]	Description	Remark
000	1 tCK.	(reset value)
001	2 tCK.	
010	3 tCK.	
011	4 tCK.	
100	5 tCK.	
101	6 tCK.	
110	7 tCK.	
111	8 tCK.	

tRFC: AUTO-REFRESH command period.

tRFC defines the minimum delay after an AUTO-REFRESH command.

During tRFC period, no command can be issued to DDR memory.

Delay Time = $2 * (tRFC + 1)$.

Bit [29:24]	Description	Remark
000000	1 tCK.	(reset value)
000001	3 tCK.	
000010	5 tCK.	
000011	7 tCK.	
... $2 * (tRFC + 1)$...	
111101	125 tCK.	
111110	127 tCK.	
111111	129 tCK.	

* tCK – one DDR memory clock cycle, typical tCK value is 7.5 ns (133MHz clock).

tMINSR: Minimum Self-Refresh / Deep-Power-Down time.

After DDR memory turns into Self-Refresh or Deep-Power-Down mode, it will NOT exit until tMINSR condition meets.

Delay Time = $tMINSR * 8 + 1$.

Bit [11:8]	Description	Remark
0000	1*8 + 1 tCK.	(reset value)
0001	2*8 + 1 tCK.	
0010	3*8 + 1 tCK.	
0011	4*8 + 1 tCK.	
... tMINSR * 8 + 1 ...	
1101	14*8 + 1 tCK.	
1110	15*8 + 1 tCK.	
1111	16*8 + 1 tCK.	

tXP: EXIT-POWER-DOWN to next valid command period.

tXP defines the EXIT-POWER-DOWN to next valid command period to all banks.

Bit [6:4]	Description	Remark
000	1 tCK.	(reset value)
001	1 tCK.	
010	2 tCK.	
011	3 tCK.	
100	4 tCK.	
101	5 tCK.	
110	6 tCK.	
111	7 tCK.	

tMRD: Load-Mode-Register to next valid command period.

tMRD defines the Load-Mode-Register to next valid command period.

Bit [1:0]	Description	Remark
00	1 tCK.	(reset value)
01	2 tCK.	
10	3 tCK.	
11	4 tCK.	

15.2.6 DREFCNT (DDR Auto-Refresh Counter)

DREFCNT																0x13010018																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								CON								CNT				Reserved				CLK_DIV	REF_EN						
RST	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31~24, 7-4: Reserved. Writing has no effect, read as zero.

CON: A constant value used to compare with the CNT value.

After reset, CON=0xFF and CNT=0x00;

It is not recommended to set CON=0x00.

CNT: 8-bit counter; When the value of CNT match the value of CON, flag bit EQU is set high

and an auto-refresh command will be issued to DDR memory. READ only.

CLK_DIV : Clock Divider.

Divide the dclk to generate a lower frequency of clock to drive the auto-refresh counter. This helps to save power consumption.

Set CLK_DIV=0 can disable the clock to auto-refresh counter.

When the DDR memory is in self-refresh mode or in deep-power-down mode, disable the clock of auto-refresh counter can save power consumption.

Future more, the module clock to DDRC can also be stopped.

dclk is CKO clock, When ddr work in 500Mbps, dclk is 250Mhz.

Bit [3:1]	Description	Remark
000	dclk / 16.	(reset value)
001	dclk / 32.	
010	dclk / 64.	
011	dclk / 128.	
100	dclk / 256.	
101	dclk / 512.	
110	dclk / 1024.	
111	--	

REF_EN: Enable Refresh Counter.

Software set REF_EN=1 right after initialize ddr memory.

Bit [29]	Description	Remark
0	Enable auto-refresh counter.	(reset value)
1	Disable auto-refresh counter.	

15.2.7 DMMAP0,1 (DDR Memory Map Configure Register)

The physical base address and size of external DDR Memory can be configured by DMMAP register.

The size of external DDR Memory must be: $2^{(24+n)}$, n=0, 1, 2, 3,

When the following equation is met:

(AXI_BUS_Address[31:24] & MASK[7:0]) == BASE

The DDR Memory is selected.

DMMAP0,1																0x13010024, 0x13010028																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																BASE								MASK							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bits 31~16: Reserved. Writing has no effect, read as zero.

BASE: base address.

MASK: address mask.

Examples:

- 1 DDR address space in system memory : 0x2000_0000 ~ 0x2FFF_FFFF (256MB)
BASE=0x20 MASK=0xF0.
- 2 DDR address space in system memory : 0x5000_0000 ~ 0x57FF_FFFF (128MB)
BASE=0x50 MASK=0xF8.

NOTE: If DDRC is disabled, please set DMMAP=0x0000_FF00 (reset value).

15.2.8 DDLP (DDR DFI low power handshake control register)

DDLP																0x130100BC																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													Only	UAR	UZQ	SLP				FLP				Reserved				FSR	FPD	LPEN	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1

Bits 31~19: Reserved. Writing has no effect, read as zero.

Bits 18~16: Inner usage, keep to 0

LPEN: DFI low power(lp) interface enable

FPD: 1: Use fast lp handshake process during power down.
0: Use slow lp handshake process during power down.

FSR: 1: Use fast lp handshake process during self-refresh.
0: Use slow lp handshake process during self-refresh.

SLP, FLP: slow LP handshake and fast LP handshake timing register, Just keep it as reset value.

15.2.9 DREMAP1,2,3,4,5 (DDR Address Remapping Register 1,2,3,4,5)

DREMAP1																0x1301009C																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			BIT3MP				Reserved			BIT2MP				Reserved			BIT1MP				Reserved			BIT0MP							
RST	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

DREMAP2																0x130100A0																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			BIT7MP				Reserved			BIT6MP				Reserved			BIT5MP				Reserved			BIT4MP							
RST	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

DREMAP3																0x130100A4																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			BIT11MP				Reserved			BIT10MP				Reserved			BIT9MP				Reserved			BIT8MP							
RST	0	0	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0

DREMAP4																0x130100A8																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			BIT15MP				Reserved			BIT14MP				Reserved			BIT13MP				Reserved			BIT12MP							
RST	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0

DREMAP5																0x130100AC																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			BIT19MP				Reserved			BIT18MP				Reserved			BIT17MP				Reserved			BIT16MP							
RST	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0

DREMAP1~5 are used to define address mapping in DDRC. The “BITnMP” represents address[31:20]. The low 4KB(bit [11:0]) address are reserved without any change. The high 20 bits are configurable. For example, If you want to remap address between address[27:25] and address[14:12], you can set BIT12MP as 0, BIT13MP as 1, BIT14MP as 2, BIT0MP as 12, BIT1MP

as 13 and BIT2MP as 14.

Original address:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11~0
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Reserved

Remapped address

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11~0
19	18	17	16	2	1	0	12	11	10	9	8	7	6	5	4	3	15	14	13	Reserved

15.2.10 WCMDCTRL1 (Performance wcmd reorder & grouping)

Terminology:

- Priority: priority of command, the command with higher priority will be processed prior to that with lower priority. The priority is generated from:
 1. on-port control signals (hpri, ARPRI, AWPRI)
 2. iport_pri (register configurable for each port)
 3. QoS counter
- Page-hit-conflict:
 - Page-hit: wpage-hit
 - Page-conflict: the page of write command is conflict with the page of read command. In this case, to guarantee the read-write coherency, the write command must be process prior to read command.

WCMDCTRL1																0x13010100																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	wcmd_lock_thd				wpage_hit_max								pend_qos_cnt								wram_in_use				disable_reorder	msk_empty_wfifo						
RST	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bits	Name	Description	Reset
[31:27]	wcmd_lock_thd	Indicates how many wcmd(write commands) are grouped together. Higher value improves total bandwidth of ddr, but leads to longer read command latency.	0x8
[26:19]	wpage_hit_max	Wpage-hit affects the reorder of wcmd, the wcmd with same page should be bonded for higher performance. This register indicates the maximum number of wcmd to be	0x8

		bonded. Higher value improves total bandwidth of ddr, but leads to longer read maximum latency.	
[18:7]	pend_qos_cnt	Once the wcmd is accepted, a counter indicates how long the command is pending, when the pending time is exceed Pend_QoS once, the priority of this command will increase 1, up to 3, which increases the priority to be processed. This mechanism avoids the wcmd with lower priority pending too long. Range: 0x001 ~0xFFF NOTE: configure step: <ol style="list-style-type: none"> 1. set <i>disable_reorder</i> to 0. 2. execute <i>step1</i> for 14 times 3. set <i>wcmd_pend_qos</i> to expect value 4. set <i>disable_reorder</i> to 1, (Optional) Higher value improves total bandwidth of ddr, but leads to longer maximum latency for write command with lower priority. Dynamic configuration is not supported.	0x100
[6:2]	wram_in_use	Reserved, Do Not change this value	0x10
[1]	disable_reorder	0: wcmd will be reordered based on priority and page-hit-conflict 1: wcmd will not be reordered, the commands will be process with the order of being received. Dynamic configuration is not recommended.	0
[0]	msk_empty_wfifo	0: wcmd is involved in arbitration immediately. 1: wcmd is not involved in arbitration until receives enough wdata.	0

15.2.11 RCMDCTRL0 (Performance rcmd request control)

Terminology:

rfifo_thd: read fifo threshold. Read command is not involved in arbitration until the vacancy in rfifo is higher than rfifo threshold. This should improves the ddr performance when processing accessing discrete pages. However, for page-access (commands access the same page continuously) the performance will be lagged as the threshold affects the command grouping.

rcmd_igr_cflit: guarantee coherency between wcmd and rcmd. If the coherency if not required in some application (such as LCD), ignoring of confliction would improve ddr read performance.

RCMDCTRL0																0x13010104																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ch3_rcmd_igr_cflit	ch3_rfifo_thd_en	ch3_rfifo_thd						ch2_rcmd_igr_cflit	ch2_rfifo_thd_en	ch2_rfifo_thd						ch1_rcmd_igr_cflit	ch1_rfifo_thd_en	ch1_rfifo_thd						ch0_rcmd_igr_cflit	ch0_rfifo_thd_en	ch0_rfifo_thd					
RST	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0

Bits	Name	Description	Reset
[31]	ch3_rcmd_igr_cflit	1: enable. Consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x0
[30]	ch3_fifo_thd_en	0: disable rfifo_thd (recommanded) 1: enable rfifo_thd	0x0
[29:24]	ch3_rfifo_thd	If Ch3 rfifo level threshold higher than this value, rcmd request is masked	0x8
[23]	ch2_rcmd_igr_cflit	1: enable. Consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x0
[22]	ch2_fifo_thd_en	0: disable rfifo_thd (recommanded) 1: enable rfifo_thd	0x0
[21:16]	ch2_rfifo_thd	If Ch3 rfifo level threshold higher than this value, rcmd request is masked	0x8
[15]	ch1_rcmd_igr_cflit	1: enable. Consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x0
[14]	ch1_fifo_thd_en	0: disable rfifo_thd (recommanded) 1: enable rfifo_thd	0x1
[13:8]	ch1_rfifo_thd	If Ch3 rfifo level threshold higher than this value, rcmd request is masked	0x8
[7]	ch0_rcmd_igr_cflit	1: enable. Consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x0
[6]	ch0_fifo_thd_en	0: disable rfifo_thd (recommanded) 1: enable rfifo_thd	0x1
[5:0]	ch0_rfifo_thd	If Ch3 rfifo level threshold higher than this value, rcmd request is masked	0x8

15.2.12 RCMDCTRL1 (Performance rcmd request control)

RCMDCTRL1															0x13010108																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
																Ch5_rcmd_igr_cflit	Ch5_rfifo_thd_en	Ch5_rfifo_thd					Ch4_rcmd_igr_cflit	Ch4_rfifo_thd_en	Ch4_rfifo_thd								
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Bits	Name	Description	Reset
[15]	ch5_rcmd_igr_cflit	0: disable rfifo_thd (recommended) 1: enable rfifo_thd	0x0
[14]	ch5_fifo_thd_en	1: enable. consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x1
[13:8]	ch5_rfifo_thd	If Ch1 rfifo level threshold higher than this value, rcmd request is masked	0x8
[7]	ch4_rcmd_igr_cflit	0: disable rfifo_thd (recommended) 1: enable rfifo_thd	0x0
[6]	ch4_fifo_thd_en	1: enable. consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x0
[5:0]	ch4_rfifo_thd	If Ch1 rfifo level threshold higher than this value, rcmd request is masked	0x8

15.2.13 WDATTHD0 (performance wcmd request control)

Terminology:

Wfifo_thd: wdata fifo threshold. In axi port, as the wdata might be far behind of wcmd. The threshold guarantees the wcmd will not be involved in arbitration until receive enough wdata. This improves ddr performance for axi write access which wdata does not continuously transfer on bus. But for continuous transfer, the performance would be worse.

WDATTHD0															0x13010114																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved	Ch3_wfifo_thd_en	Ch3_wfifo_thd					Reserved	Ch2_wfifo_thd_en	Ch2_wfifo_thd					Reserved	Ch1_wfifo_thd_en	Ch1_wfifo_thd					Reserved	Ch0_wfifo_thd_en	Ch0_wfifo_thd								
RST	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	0	0

Bits	Name	Description	RW
30	Ch3_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x0
[29:24]	Ch3_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8
[22]	Ch2_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x0
[21:16]	Ch2_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8
[14]	Ch1_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x1
[13:8]	Ch1_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8
[6]	Ch0_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x1
[5:0]	Ch0_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8

15.2.14 WDATTHD1 (performance wcmd request control)

WDATTHD1																0x13010118																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															Ch5_wfifo_thd_en	Ch5_wfifo_thd					Reserved	Ch4_wfifo_thd_en	Ch4_wfifo_thd								
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0

Bits	Name	Description	RW
[14]	Ch5_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x1
[13:8]	Ch5_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8
[6]	Ch4_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x1
[5:0]	Ch4_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8

15.2.15 IPORTPRI (performance priority control)

Terminology:

Priority affects on arbitration between ports. Transfers with higher priority will be grant first

Port_pri: priority provided by masters.

Iport_pri: register configurable which applies to each ports. The final priority is the larger value of port_pri and iport_pri.

IPORTPRI																0x13010128																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	Reserved								Ch5_iport_pri_en	Reserved	Ch5_iport_pri	Ch4_iport_pri_en	Reserved	Ch4_iport_pri	Ch3_iport_pri_en	Reserved	Ch3_iport_pri	Ch2_iport_pri_en	Reserved	Ch2_iport_pri	Ch1_iport_pri_en	Reserved	Ch1_iport_pri	Ch0_iport_pri_en	Reserved	Ch0_iport_pri								
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bits	Name	Description	RW
[23]	Ch5_iport_pri_en	Enable iport_pri 0: disable, use external priority 1: enable, use the max priority of external and internal priority	0x0
[21:20]	Ch5_iport_pri	Priority of channel 0x3: the highest priority ... 0x0: the lowest priority	0x0
[19]	Ch4_iport_pri_en	Enable iport_pri 0: disable, use external priority 1: enable, use the max priority of external and internal priority	0x0
[17:16]	Ch4_iport_pri	Priority of channel 0x3: the highest priority ... 0x0: the lowest priority	0x0
[15]	Ch3_iport_pri_en	Enable iport_pri 0: disable, use external priority 1: enable, use the max priority of external and internal priority	0x0
[13:12]	Ch3_iport_pri	Priority of channel 0x3: the highest priority ... 0x0: the lowest priority	0x0
[11]	Ch2_iport_pri_en	Enable iport_pri 0: disable, use external priority 1: enable, use the max priority of external and internal priority	0x0
[9:8]	Ch2_iport_pri	Priority of channel 0x3: the highest priority ...	0x0

		0x0: the lowest priority	
[7]	Ch1_iport_pri_en	Enable iport_pri 0: disable, use external priority 1: enable, use the max priority of external and internal priority	0x0
[5:4]	Ch1_iport_pri	Priority of channel 0x3: the highest priority ... 0x0: the lowest priority	0x0
[3]	Ch0_iport_pri_en	Enable iport_pri 0: disable, use external priority 1: enable, use the max priority of external and internal priority	0x0
[1:0]	Ch0_iport_pri	Priority of channel 0x3: the highest priority ... 0x0: the lowest priority	0x0

15.2.16 CHQOS0,1,2,3,4,5 (performance QoS control)

Terminology:

Qos: Generally speaking, the Qos controller is a timer, it indicates the how long the command is pending. Once the timer exceed predefined threshold (con), the priority increases by 1, up to top-limit (max). The higher priority is, the sooner it will be processed. This register affects the latency.

CHQOS0,1,2,3,4,5,6																0x1301007c,80,84,88,8c,90,94																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Qos_en		Outstanding_en		Reserved												max		con														
RST	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	Name	Description	Rrset
[31]	Qos_en	enable	0x1
[17:16]	Max	Up-limit of priority	0x3
[15:0]	Con	Threshold of timer	0x100

15.2.17 CPM_DRCG

CPM_DRCG																0xB0000D0																		
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	Reserved																														DLLRESET	CFGCLKEN		
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	Name	Description	Rrset
[31:2]	Reserved	Inner usage, keep in default value	0x0
[1]	DLLRESET	1: reset the DLL in DDR PHY 0: disable the reset.	0x0
[0]	CFGCLKEN	Inner usage, keep to 1	0x1

15.3 Functional Description

15.3.1 DDRC and DDR2 Memory Initialization Sequence

The following content is just an example for initialization, the sequence is already implementation in our BSP.

15.3.1.1 Example

One 512Mb x16 DDR2 device connected on CS0;

No memory device connected on CS1;

DCK = 133 MHz, CL = 3.

- 1 After system reset, wait system clock stable before initialize ddr.
 - 2 Configure the Clock-Control module for ddr clocks.
 - 3 DDR Memory device need at least 200us initialization time after power-on before it can accept any command.
- ```
//-----
// INIT DDRC
//-----
4 Configure DCFG = 0x.
5 Configure DTIMING1~6 = 0x.
6 Configure DMAP0 = 0x.
7 Configure DMAP1 = 0x0000FF00.
//-----
// INIT DDR PHY
//-----
8 Configure DDR PHY and finish PHY training process.(relate to DDRPHY spec)
```

```

//-----
// DDR performance control configure (optional)
//-----
9 Configure RCMDCTRL0 = 0x08080808 (optional)
10 Configure RCMDCTRL1 = 0x00000808 (optional)
11 Configure WDATTHD0 = 0x08080808 (optional)
12 Configure WDATTHD1 = 0x00000808 (optional)
//-----
// INIT DDR memory device
//-----
13 Set CKE Pin HIGH : Configure DCTRL = 0x00000002.
//-----
// Enable Refresh Counter
//-----
14 Enable Refresh Counter : Configure DREFCNT = 0x.
15 AUTO-REFRESH : Configure DCTRL = 0x.
//-----
// END INITIALIZING SEQUENCE
//-----

```

## 15.4 Change Clock Frequency

To reduce power consumption, the system clock frequency may be changed frequently according to the application. There are 3 ways to change the clock frequency.

### 15.4.1 Manually SELF-REFRESH Mode

DDR can stay in SELF-REFRESH & DEEP-POWER-DOWN mode for a long period of time. System clock frequency can be changed during this time. Even more, the clocks to DDRC module can also be stopped to save power-consumption.

Reference Sequence:

- 1 Manually issue SELF-REFRESH command to DDR.
- 2 Change relative register in CPM.
- 3 Change system clock frequency.
- 4 Drive DDR exit SELF-REFRESH mode.

### 15.4.2 CPM driven SELF-REFRESH Mode

CPM will auto drive DDRC to self-refresh mode, when use ddr2, ddr3.

To change clock frequency, please refer CPM spec.

### 15.4.3 DLL bypass mode

The clock frequency on which DLL in ddr phy works much faster than 200MHz, As the result, when ddr clock is failed to meet this condition (for example, work in 12MHz) , the DLL must be bypassed in order to supply clock to ddr controller.

Step1, Configure DLL into reset mode to guarantee a driving clock.

```
(0xB00000D0) CPM_DRCG[1] <= 1;
```

Step2, Configure DLL into bypass mode.

```
(0xB3011014) DDR_ACDLLCR[0] <= 1;
```

Then, (0xB00000D0) CPM\_DRCG[1] <= 0;

### 15.5 Data Endian

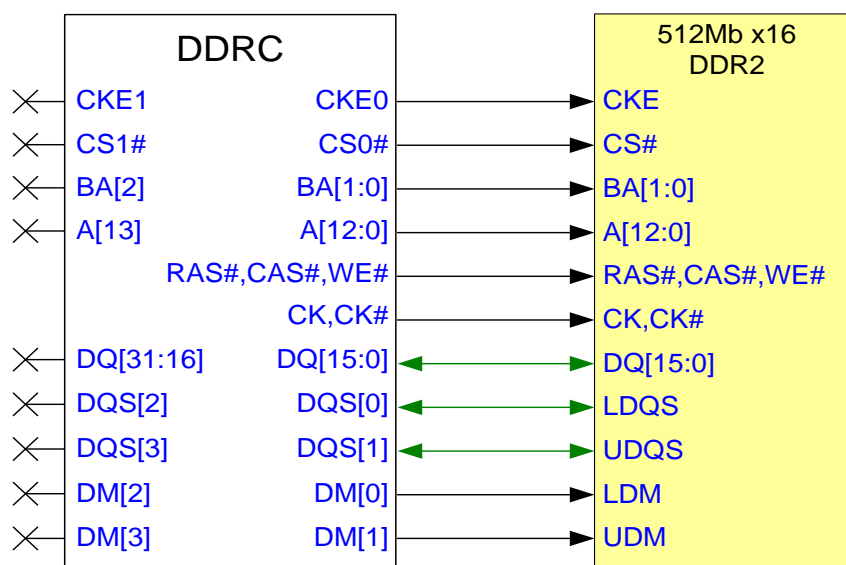
Fix to little Endian.

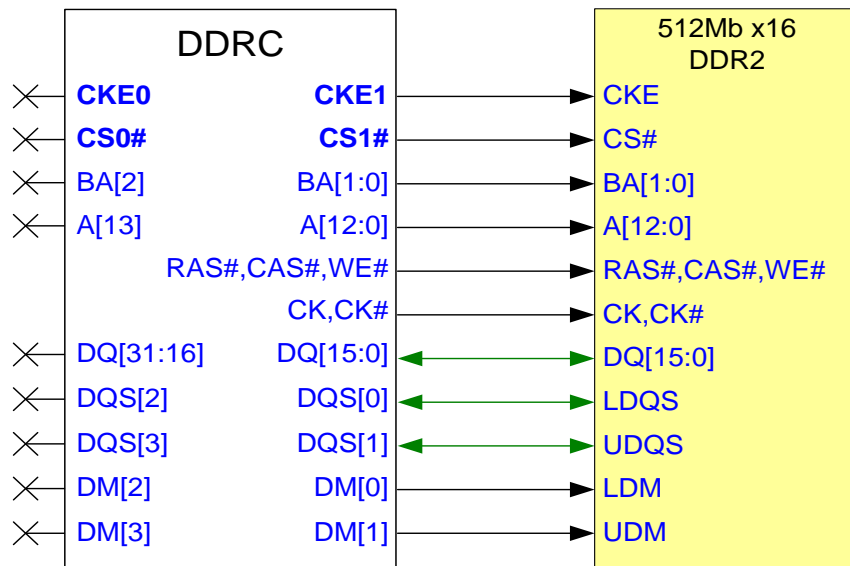
### 15.6 DDR Connection Diagrams

The following diagrams give examples on the connection to external DDR2 devices.

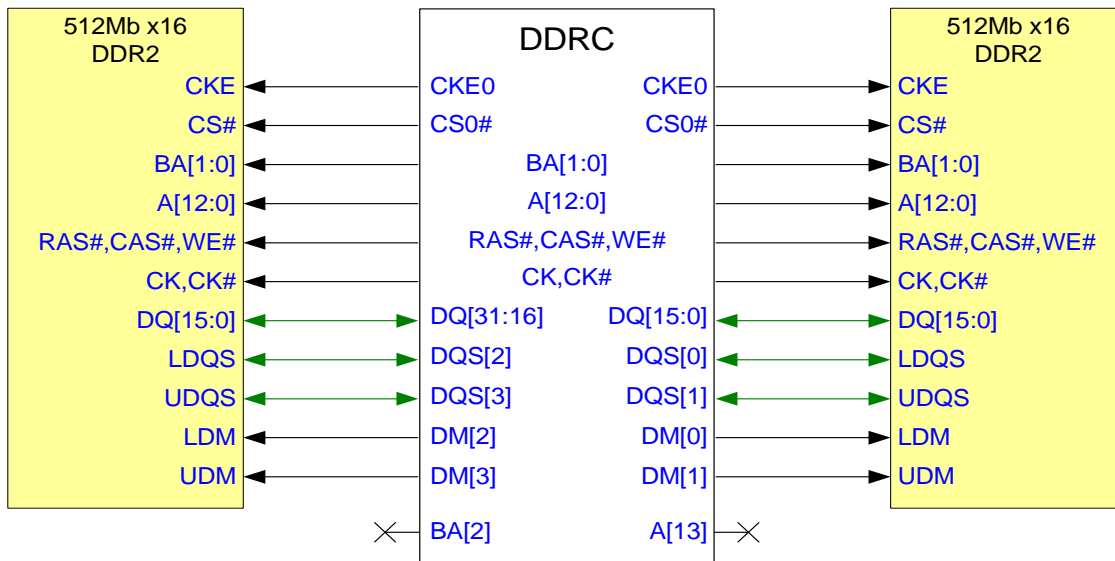
Note not all the possible connections are listed.

#### 15.6.1 Connection to one 512Mb x16 DDR2 device





### 15.6.2 Connection to two 512Mb x16 DDR2 devices



# 16 External NAND Memory Controller

## 16.1 Overview

The External NAND Memory Controller (NEMC) divides the off-chip memory space and outputs control signals complying with specifications of various types of static memory and bus interfaces. It enables the connection of static memory such as conventional NAND flash memory, Toggle NAND flash memory, etc. to this processor.

- Static memory interface
  - Support 6 external chip selection CS6~1#. Each bank can be configured separately
  - The size and base address of static memory banks are programmable
  - Direct interface to 8-bit bus width external memory interface devices or external static memory to each bank. Read/Write strobe setup time and hold time periods can be programmed and inserted in an access cycle to enable connection to low-speed memory
  - Wait insertion by WAIT pin
  - Automatic wait cycle insertion to prevent data bus collisions in case of consecutive memory accesses to different banks, or a read access followed by a write access to the same bank
- NAND flash interface
  - Support on CS6~CS1, sharing with static memory bank6~bank1
  - Support both of conventional NAND flash memory and Toggle NAND flash memory
  - Support most types of NAND flashes, 8-bit data access, 512B/2K/4K/8KB page size. For 512B page size, 3 and 4 address cycles are supported. For 2K/4K/8KB page size, 4 and 5 address cycles are supported
  - Support read/erase/program NAND flash memory
  - Support boot from NAND flash

## 16.2 Pin Description

Following table list the NEMC pins.

**Table 16-1 NEMC Pin Description**

| Pin Name                 | I/O | Signal    | Description                                                       |
|--------------------------|-----|-----------|-------------------------------------------------------------------|
| Data Bus                 | I/O | SD7 – SD0 | Data I/O.                                                         |
| Address bus              | O   | SA5–SA0   | Address output.                                                   |
| Static chip select 6 ~ 1 | O   | CS6~1#    | Chip select signal that indicates the static bank being accessed. |
| Read enable              | O   | RD# /     | For Static memory read enable signal.                             |
| Write enable             | O   | WE# /     | Static memory write enable signal.                                |

|                               |     |         |                                                                                                         |
|-------------------------------|-----|---------|---------------------------------------------------------------------------------------------------------|
| Wait                          | I   | Wait# / | External wait state request signal for memory-like devices.                                             |
| NAND flash read enable        | O   | FRE#    | NAND flash read enable signal.                                                                          |
| NAND flash write enable       | O   | FWE#    | NAND flash write enable signal.                                                                         |
| NAND flash ready/busy         | I   | FRB#    | Indicates NAND flash is ready or busy. (When Nand flash boot, GPC30 is used as FRB# of CS1#)            |
| Toggle NAND flash data strobe | I/O | DQS     | Toggle NAND data output is aligned with DQS falling/rising edge, and data inputs at DQS falling/rising. |

### 16.3 Physical Address Space Map

Both virtual spaces and physical spaces are 32-bit wide in this architecture. Virtual addresses are translated by MMU into physical address which is further divided into several partitions for static memory, SDRAM, and internal I/O.

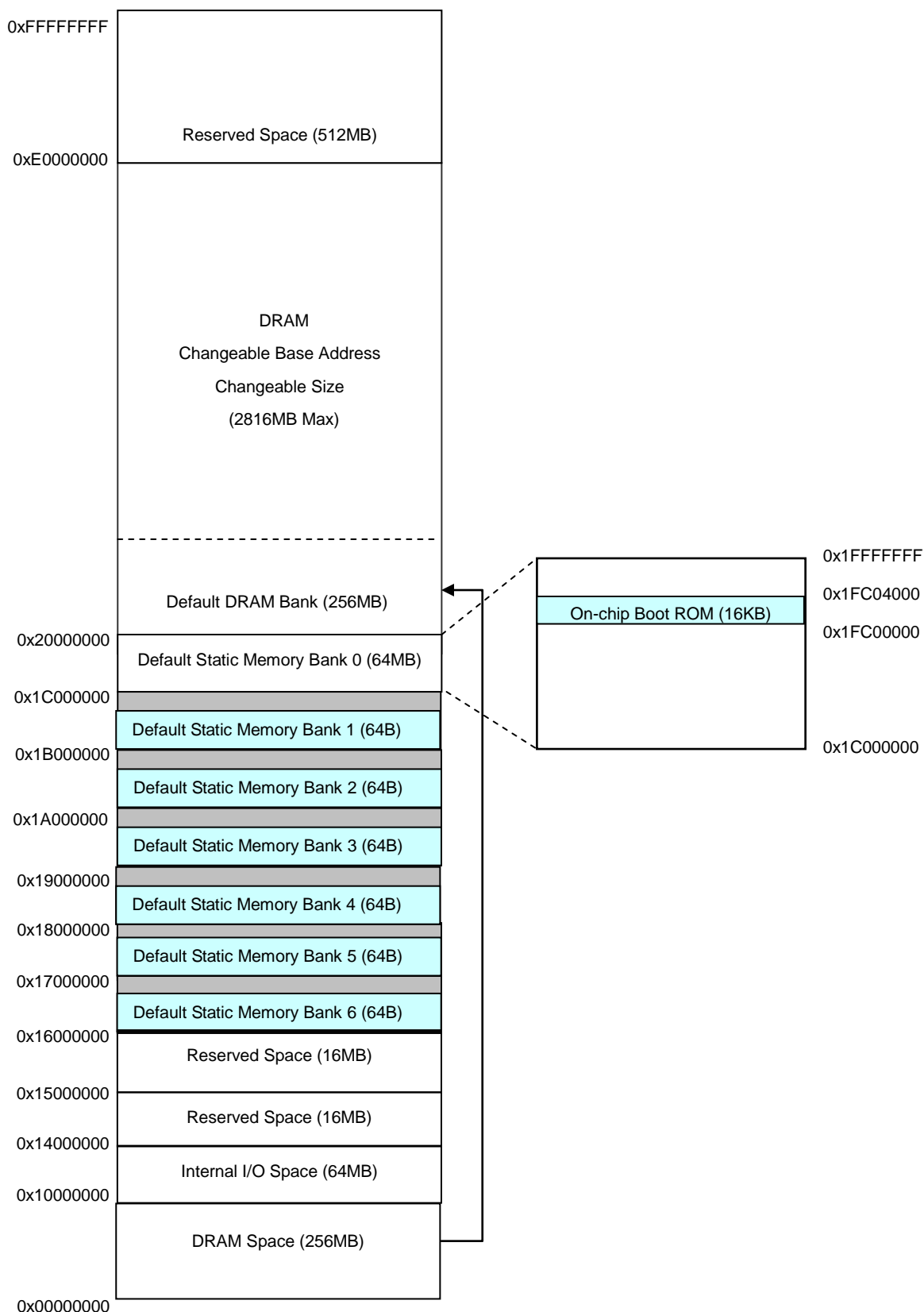


Figure 16-1 Physical Address Space Map



**Table 16-2 Physical Address Space Map**

| Start Address | End Address | Connectable Memory         | Capacity |
|---------------|-------------|----------------------------|----------|
| 0x00000000    | 0x0FFFFFFF  | SDRAM Memory               | 256MB    |
| 0x10000000    | 0x13FFFFFF  | Internal I/O Devices       | 64MB     |
| 0x14000000    | 0x14FFFFFF  | Reserved                   | 16MB     |
| 0x15000000    | 0x15FFFFFF  | Reserved                   | 16MB     |
| 0x16000000    | 0x1600003F  | Static Memory, CS6#        | 64B      |
| 0x16000040    | 0x16FFFFFF  | Reserved                   |          |
| 0x17000000    | 0x1700003F  | Static Memory, CS5#        | 64B      |
| 0x17000040    | 0x17FFFFFF  | Reserved                   |          |
| 0x18000000    | 0x1800003F  | Static Memory, CS4#        | 64B      |
| 0x18000040    | 0x18FFFFFF  | Reserved                   |          |
| 0x19000000    | 0x1900003F  | Static Memory, CS3#        | 64B      |
| 0x19000040    | 0x19FFFFFF  | Reserved                   |          |
| 0x1A000000    | 0x1A00003F  | Static Memory, CS2#        | 64B      |
| 0x1A000040    | 0x1AFFFFFF  | Reserved                   |          |
| 0x1B000000    | 0x1B00003F  | Static Memory, CS1#        | 64B      |
| 0x1B000040    | 0x1BFFFFFF  | Reserved                   |          |
| 0x1C000000    | 0x1FBFFFFF  | Reserved                   |          |
| 0x1FC00000    | 0x1FC03FFF  | On-chip Boot ROM<br>(16kB) | 16KB     |
| 0x1FC04000    | 0x1FFFFFFF  | Reserved                   |          |
| 0x20000000    | 0xDFFFFFFF  | SDRAM Memory               | 3072MB   |
| 0xE0000000    | 0xFFFFFFFF  | Reserved                   | 512MB    |

The base address and size of each memory banks are configurable. Software can re-configure these memory banks according to the actual connected memories. Following table lists the default configuration after reset.

**Table 16-3 Default Configuration of NEMC Chip Select Signals**

| Chip-Select Signal | Connected Memory     | Capacity | Memory Width <sup>*1</sup> | Start Address | End Address |
|--------------------|----------------------|----------|----------------------------|---------------|-------------|
| CS1#               | Static memory bank 1 | 64 B     | 8, 32                      | 0x1B000000    | 0x1B00003F  |
| CS2#               | Static memory bank 2 | 64 B     | 8, 32                      | 0x1A000000    | 0x1A00003F  |
| CS3#               | Static memory bank 3 | 64 B     | 8, 32                      | 0x19000000    | 0x1900003F  |
| CS4#               | Static memory bank 4 | 64 B     | 8, 32                      | 0x18000000    | 0x1800003F  |
| CS5#               | Static memory bank 5 | 64 B     | 8, 32                      | 0x17000000    | 0x1700003F  |
| CS6#               | Static memory bank 6 | 64 B     | 8, 32                      | 0x16000000    | 0x1600003F  |

**NOTES:**

- 1 The 16KB address space from H'1FC00000 to H'1FC03FFF in bank 0 is mapped to on-chip boot ROM. The other memory spaces in bank 0 are not used.

## 16.4 Static Memory Interface

NEMC provides a glueless interface to normal static memory which don't need byte control like SRAM, memory interface IO devices, etc.. It can directly control up to 6 devices using six chip select lines. Additional devices may be supported through external decoding of the address bus.

Each chip select can directly access memory or IO devices that are 8-bits wide. Each device connected to a chip select line has 2 associated registers that control its operation and the access timing to the external device. The Static Memory Control Register SMCRn specifies various configurations for the device. The Static Memory Address Configuration Register SACRn specifies the base address and size for each device, enabling any device to be located anywhere in the physical address range.

The static memory interface includes the following signals:

- Six chip selects, CS6~1#
- Six address signals, SA5-SA0
- One read enable, RD#
- One write enable, WE#
- One wait pin, WAIT#

The SMT field in SMCRn registers specifies the type of memory and BW field specifies the bus width. BOOT\_SEL[1:0] pin defines whether system boot from Nor or Nand flash and the page size when boot from Nand flash.

### 16.4.1 Register Description

**Table 16-4 Static Memory Interface Registers**

| Name  | Description                                         | RW | Reset Value | Address    | Access Width |
|-------|-----------------------------------------------------|----|-------------|------------|--------------|
| SMCR1 | Static memory control register 1                    | RW | 0x0FFF7700  | 0x13410014 | 32           |
| SMCR2 | Static memory control register 2                    | RW | 0x0FFF7700  | 0x13410018 | 32           |
| SMCR3 | Static memory control register 3                    | RW | 0x0FFF7700  | 0x1341001C | 32           |
| SMCR4 | Static memory control register 4                    | RW | 0x0FFF7700  | 0x13410020 | 32           |
| SMCR5 | Static memory control register 5                    | RW | 0x0FFF7700  | 0x13410024 | 32           |
| SMCR6 | Static memory control register 6                    | RW | 0x0FFF7700  | 0x13410028 | 32           |
| SACR1 | Static memory bank 1 address configuration register | RW | 0x00001AFE  | 0x13410034 | 32           |
| SACR2 | Static memory bank 2 address configuration register | RW | 0x000018FE  | 0x13410038 | 32           |
| SACR3 | Static memory bank 3 address configuration register | RW | 0x000017FF  | 0x1341003C | 32           |

|       |                                                     |    |            |            |    |
|-------|-----------------------------------------------------|----|------------|------------|----|
| SACR4 | Static memory bank 4 address configuration register | RW | 0x000016FF | 0x13410040 | 32 |
| SACR5 | Static memory bank 5 address configuration register | RW | 0x000015FF | 0x13410044 | 32 |
| SACR6 | Static memory bank 6 address configuration register | RW | 0x000014FF | 0x13410048 | 32 |

### 16.4.1.1 Static Memory Control Register (SMCR1~6)

SMCR1~6 are 32-bit read/write registers that contain control bits for static memory. On reset, SMCR1~6 are initialized to 0x0FFF7700.

NOTE. If use the conventional type NAND (i.e. SDR NAND), all of the Command Latch Cycle/Address Latch Cycle/Data input Timing/Data output Timing/Read ID Operation/Read Status Cycle will be configured by SMCRn.

But, if use the high-speed Toggle DDR NAND, only the Command Latch Cycle/Address Latch Cycle will be configured by SMCRn, and Data input Timing/Data output Timing/Read ID Operation/Read Status Cycle are configured by Toggle NAND Control Register(TGCRn).

|       |          |    |      |    |    |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |   |   |    |   |            |   |    |     |   |   |
|-------|----------|----|------|----|----|----|----|----|-----|----|----|----|-----|----|----|----|-----|----|----|----|-----|----|---|---|----|---|------------|---|----|-----|---|---|
| SMCR1 |          |    |      |    |    |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |   |   |    |   | 0x13410014 |   |    |     |   |   |
| SMCR2 |          |    |      |    |    |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |   |   |    |   | 0x13410018 |   |    |     |   |   |
| SMCR3 |          |    |      |    |    |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |   |   |    |   | 0x1341001C |   |    |     |   |   |
| SMCR4 |          |    |      |    |    |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |   |   |    |   | 0x13410020 |   |    |     |   |   |
| SMCR5 |          |    |      |    |    |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |   |   |    |   | 0x13410024 |   |    |     |   |   |
| SMCR6 |          |    |      |    |    |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |   |   |    |   | 0x13410028 |   |    |     |   |   |
| Bit   | 31       | 30 | 29   | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19  | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11  | 10 | 9 | 8 | 7  | 6 | 5          | 4 | 3  | 2   | 1 | 0 |
|       | Reserved |    | STRV |    |    |    |    |    | TAW |    |    |    | TBP |    |    |    | TAH |    |    |    | TAS |    |   |   | BW |   | Reserved   |   | BL | SMT |   |   |
| RST   | 0        | 0  | 0    | 0  | 1  | 1  | 1  | 1  | 1   | 1  | 1  | 1  | 1   | 1  | 1  | 1  | 0   | 1  | 1  | 1  | 0   | 1  | 1 | 1 | 0  | 0 | 0          | 0 | 0  | 0   | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                       | RW |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                              | R  |
| 29:24 | STRV     | <b>Static Memory Recovery Time.</b><br>Its value is the number of idle cycles (0~63 cycles) inserted between bus cycles when switching from one bank to another bank or between a read access to a write access in the same bank. Its initial value is 0xF (15 cycles).                                           | RW |
| 23:20 | TAW      | <b>Access Wait Time.</b><br>For normal memory, these bits specify the number of wait cycles to be inserted in read strobe time. For burst ROM, these bits specify the number of wait cycles to be inserted in first data read strobe time.<br><b>TAW3~0 Wait cycle    Wait# Pin</b><br>0000    0 cycle    Ignored | RW |

|       |     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |    |
|-------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|       |     | 0001 1 cycle Enabled<br>0010 2 cycles Enabled<br>0011 3 cycles Enabled<br>0100 4 cycles Enabled<br>0101 5 cycles Enabled<br>0110 6 cycles Enabled<br>0111 7 cycles Enabled<br>1000 8 cycles Enabled<br>1001 9 cycles Enabled<br>1010 10 cycles Enabled<br>1011 12 cycles Enabled<br>1100 15 cycles Enabled<br>1101 20 cycles Enabled<br>1110 25 cycles Enabled<br>1111 31 cycles Enabled (Initial Value)                                                                                                                                                                                                                                                                                                             |    |
| 19:16 | TBP | <p><b>Burst Pitch Time.</b><br/>For burst ROM, these bits specify the number of wait cycles to be inserted in subsequent access. For normal memory, these bits specify the number of wait cycles to be inserted in write strobe time.</p> <p><b>TBP3~0 Wait cycle Wait# Pin</b></p> 0000 0 cycle Ignored<br>0001 1 cycle Enabled<br>0010 2 cycles Enabled<br>0011 3 cycles Enabled<br>0100 4 cycles Enabled<br>0101 5 cycles Enabled<br>0110 6 cycles Enabled<br>0111 7 cycles Enabled<br>1000 8 cycles Enabled<br>1001 9 cycles Enabled<br>1010 10 cycles Enabled<br>1011 12 cycles Enabled<br>1100 15 cycles Enabled<br>1101 20 cycles Enabled<br>1110 25 cycles Enabled<br>1111 31 cycles Enabled (Initial Value) | RW |
| 15:12 | TAH | <p><b>Address Hold Time.</b><br/>These bits specify the number of wait cycles to be inserted from negation of read/write strobe to address.</p> <p><b>TAH2~0 Wait cycle</b></p> 0000 0 cycle<br>0001 1 cycle<br>0010 2 cycles                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW |

|      |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |
|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|      |          | 0011 3 cycles<br>0100 4 cycles<br>0101 5 cycles<br>0110 6 cycles<br>0111 7 cycles (Initial Value)<br>1000 8 cycles<br>1001 9 cycles<br>1010 10 cycles<br>1011 11 cycles<br>1100 12 cycles<br>1101 13 cycles<br>1110 14 cycles<br>1111 15 cycles                                                                                                                                                                                                                                                           |    |
| 11:8 | TAS      | <p><b>Address Setup Time.</b><br/>                 These bits specify the number of wait cycles (0~15 cycles) to be inserted from address to assertion of read/write strobe.</p> <p><b>TAS2~0 Wait cycle</b></p> 000 0 cycle<br>001 1 cycle<br>010 2 cycles<br>011 3 cycles<br>100 4 cycles<br>101 5 cycles<br>110 6 cycles<br>111 7 cycles (Initial Value)<br>1000 8 cycles<br>1001 9 cycles<br>1010 10 cycles<br>1011 11 cycles<br>1100 12 cycles<br>1101 13 cycles<br>1110 14 cycles<br>1111 15 cycles | RW |
| 7:6  | BW       | <p><b>Bus Width.</b><br/>                 These bits specify the bus width. This field is writeable and are initialized to 0 by a reset.</p> <p><b>BW1~0 Bus Width</b></p> 00 8 bits (Initial Value)<br>01 Reserved<br>10 Reserved<br>11 Reserved                                                                                                                                                                                                                                                         | RW |
| 5:3  | Reserved | Writing has no effect, read as zero.<br><b>NOTE:</b> Don't write Bit3 to 1.                                                                                                                                                                                                                                                                                                                                                                                                                               | R  |

|     |     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |    |
|-----|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2:1 | BL  | <p><b>Burst Length (BL1, BL0).</b><br/>When Burst ROM is connected; these bits specify the number of burst in an access. These bits are only valid when SMT is set to 1.</p> <p><b>BL1~0 Burst Length</b></p> <p>00 4 consecutive accesses. Can be used with 8-bit bus width (Initial Value)</p> <p>01 8 consecutive accesses. Can be used with 8-bit bus width</p> <p>10 16 consecutive accesses. Can only be used with 8-bit bus width</p> <p>11 32 consecutive accesses. Can only be used with 8-bit bus width</p> | RW |
| 0   | SMT | <p><b>Static Memory Type (SMT).</b><br/>This bit specifies the type of static memory.</p> <p><b>SMT Description</b></p> <p>0 Normal Memory (Initial Value)</p> <p>1 Burst ROM</p>                                                                                                                                                                                                                                                                                                                                     | RW |

### 16.4.1.2 Static Bank Address Configuration Register (SACR1~6)

SACR1~6 defines the physical address for static memory bank 1 to 6, respectively. Each register contains a base address and a mask. When the following equation is met:

$$(physical\_address [31:24] \& MASK_n) == BASE_n$$

The bank n is active. The *physical\_address* is address output on internal system bus. Static bank regions must be programmed so that each bank occupies a unique area of the physical address space. Programming overlapping bank regions will result in unpredictable error. These registers are initialized by a reset.

|      |                                                                                       |            |   |   |   |   |   |   |   |   |   |   |   |   |   |      |   |   |   |   |   |   |      |   |   |   |   |   |   |   |   |   |   |   |
|------|---------------------------------------------------------------------------------------|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|------|---|---|---|---|---|---|------|---|---|---|---|---|---|---|---|---|---|---|
|      | SACR1                                                                                 | 0x13410034 |   |   |   |   |   |   |   |   |   |   |   |   |   |      |   |   |   |   |   |   |      |   |   |   |   |   |   |   |   |   |   |   |
|      | SACR2                                                                                 | 0x13410038 |   |   |   |   |   |   |   |   |   |   |   |   |   |      |   |   |   |   |   |   |      |   |   |   |   |   |   |   |   |   |   |   |
|      | SACR3                                                                                 | 0x1341003C |   |   |   |   |   |   |   |   |   |   |   |   |   |      |   |   |   |   |   |   |      |   |   |   |   |   |   |   |   |   |   |   |
|      | SACR4                                                                                 | 0x13410040 |   |   |   |   |   |   |   |   |   |   |   |   |   |      |   |   |   |   |   |   |      |   |   |   |   |   |   |   |   |   |   |   |
|      | SACR5                                                                                 | 0x13410044 |   |   |   |   |   |   |   |   |   |   |   |   |   |      |   |   |   |   |   |   |      |   |   |   |   |   |   |   |   |   |   |   |
|      | SACR6                                                                                 | 0x13410048 |   |   |   |   |   |   |   |   |   |   |   |   |   |      |   |   |   |   |   |   |      |   |   |   |   |   |   |   |   |   |   |   |
| Bit  | 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |            |   |   |   |   |   |   |   |   |   |   |   |   |   |      |   |   |   |   |   |   |      |   |   |   |   |   |   |   |   |   |   |   |
|      | Reserved                                                                              |            |   |   |   |   |   |   |   |   |   |   |   |   |   | BASE |   |   |   |   |   |   | MASK |   |   |   |   |   |   |   |   |   |   |   |
| RST1 | 0                                                                                     | 0          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0 | 1 | 1 | 0 | 1 | 1    | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| RST2 | 0                                                                                     | 0          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0 | 1 | 1 | 0 | 1 | 0    | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

|      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RST3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| RST4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| RST5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |   |
| RST6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |   |

| Bits  | Name     | Description                                                                                                                                                                                                                                                   | RW |
|-------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:16 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                          | R  |
| 15:8  | BASE     | <b>Address Base:</b> Defines the base address of Static Bank n (n = 1 to 6).<br>The initial values are:<br>SACR1.BASE       0x1B<br>SACR2.BASE       0x1A<br>SACR3.BASE       0x19<br>SACR4.BASE       0x18<br>SACR5.BASE       0x17<br>SACR6.BASE       0x16 | RW |
| 23:20 | MASK     | <b>Address Mask:</b> Defines the mask of Static Bank n (n = 1 to 6).<br>The initial values are:<br>SACR1.MASK       0xFF<br>SACR2.MASK       0xFF<br>SACR3.MASK       0xFF<br>SACR4.MASK       0xFF<br>SACR5.MASK       0xFF<br>SACR6.MASK       0xFF         | RW |

#### 16.4.2 Example of Connection

Following figures shows examples of connection to 8-bit data width normal memory.

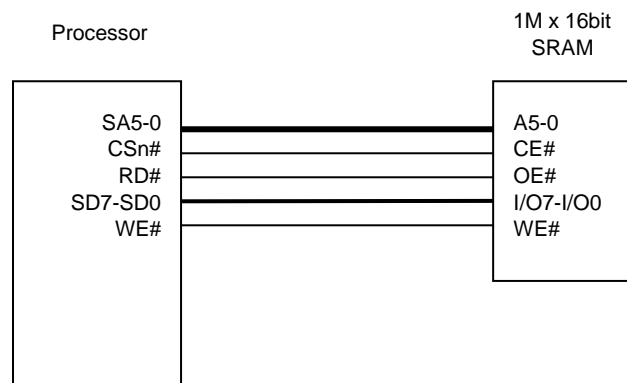
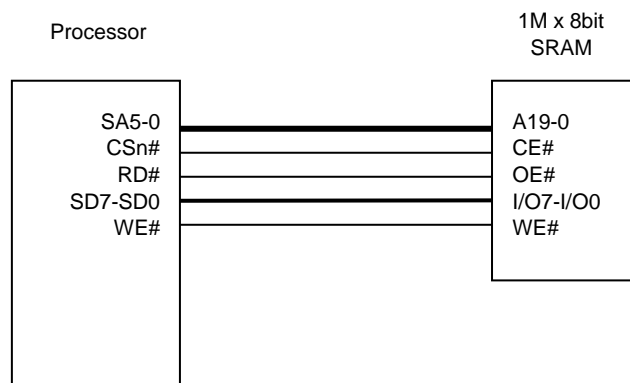


Figure 16-2 Example of 16-Bit Data Width SRAM Connection



**Figure 16-3 Example of 8-Bit Data Width SRAM Connection**

### 16.4.3 Basic Interface

When SMT field in SMCRn ( $n = 1$  to  $6$ ) is 0, normal memory (non-burst ROM, Flash, normal SRAM or memory-like device) is connected to bank  $n$ . When bank  $n$  ( $n = 1$  to  $6$ ) is accessed, CSn# is asserted as soon as address is output. In addition, the RD# signal, which can be used as OE#, and write control signals WE# is asserted.

The TAS field in SMCRn is the latency from CSn# to read/write strobe. The TAW3 field is the delay time of RD# in read access. TBP3~0 field is the delay time of WE# and WEn# in write access. In addition, any number of waits can be inserted by means of the external pin (WAIT#). The TAH field is the latency from RD# and WEn# negation to CSn# negation, also the hold time to address and write data.

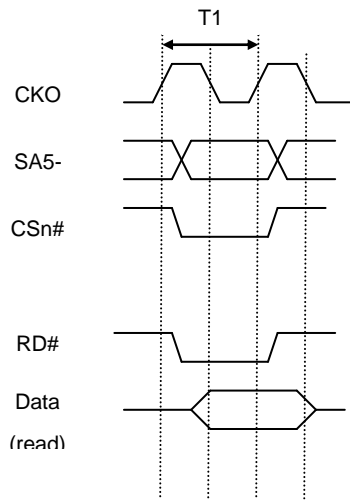
All kinds of normal memories (non-burst ROM, normal SRAM and Flash) have the same read and write timing. There are some requirements for writes to flash memory. Flash memory space must be un-cacheable and un-buffered. Writes must be exactly the width of the populated Flash devices on the data bus (no byte writes or word writes to a 16-bit bus, and so on). Software is responsible for partitioning commands and data, and writing them out to Flash in the appropriate sequence.

### Glossary

- Th – hold cycle
- Tw – wait cycle
- Ts – setup cycle
- T1 – read inherent cycle or first write inherent cycle
- T2 – last write inherent cycle
- Tb – burst read inherent cycle

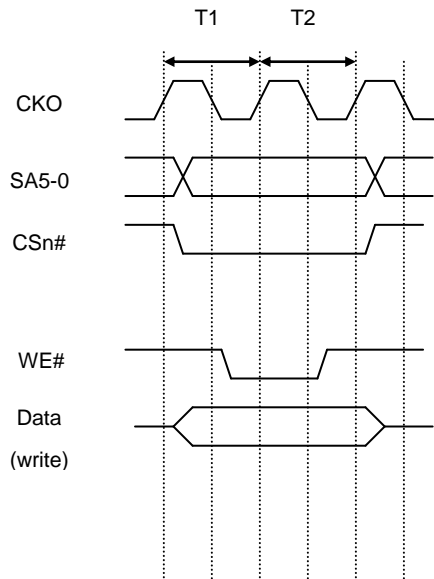
Following figures show the timing of normal memory. A no-wait read access is completed in one cycle and a no-wait write access is completed in two cycles. Therefore, there is no negation period in case of access at minimum pitch.





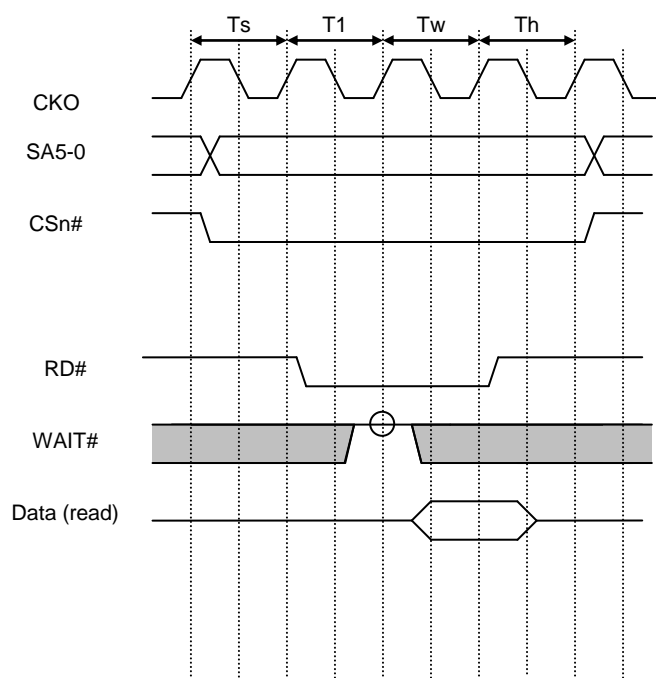
\*In this example, SMCRn:MT = 0, TAS = 0, TAW = 0, TAH = 0

Figure 16-4 Basic Timing of Normal Memory Read



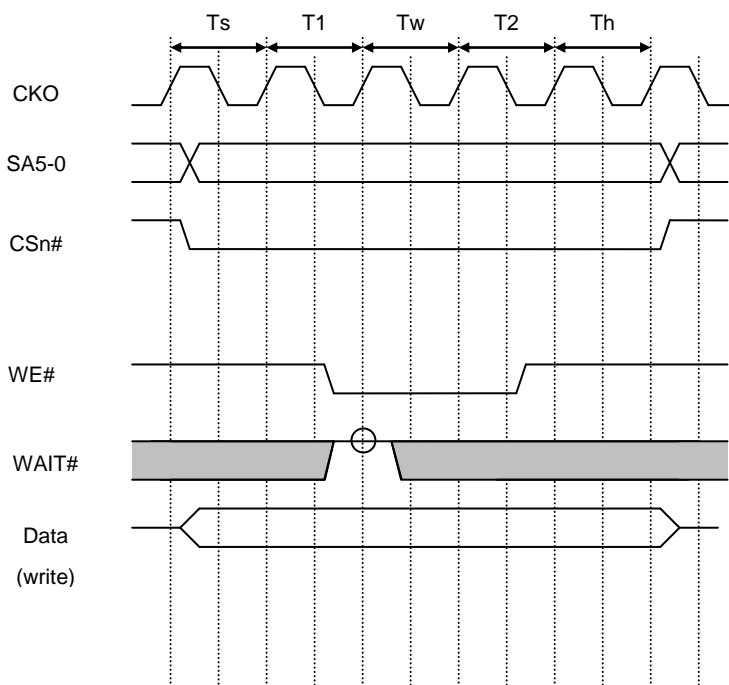
\*In this example, SMCRn: SMT = 0, TAS = 0, TBP = 0, TAH = 0

Figure 16-5 Basic Timing of Normal Memory Write



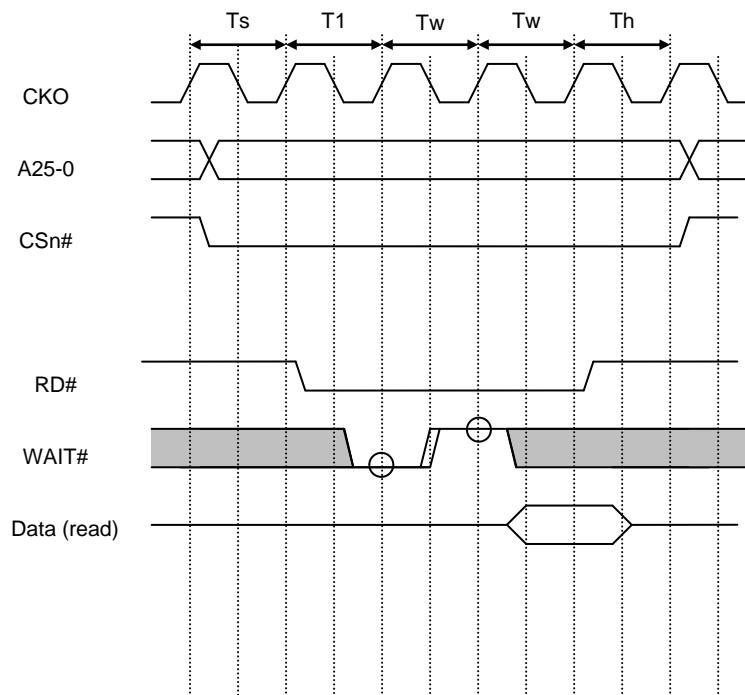
\*In this example, SMCRn: SMT = 0, TAS = 1, TAW = 1, TAH = 1

**Figure 16-6 Normal Memory Read Timing With Wait (Software Wait Only)**



\*In this example, SMCRn: SMT = 0, TAS = 1, TBP = 1, TAH = 1

**Figure 16-7 Normal Memory Write Timing With Wait (Software Wait Only)**



\*In this example, SMCRn: SMT = 0, TAS = 1, TAW = 1, TAH=1

**Figure 16-8 Normal Memory Read Timing With Wait (Wait Cycle Insertion by WAIT# pin)**

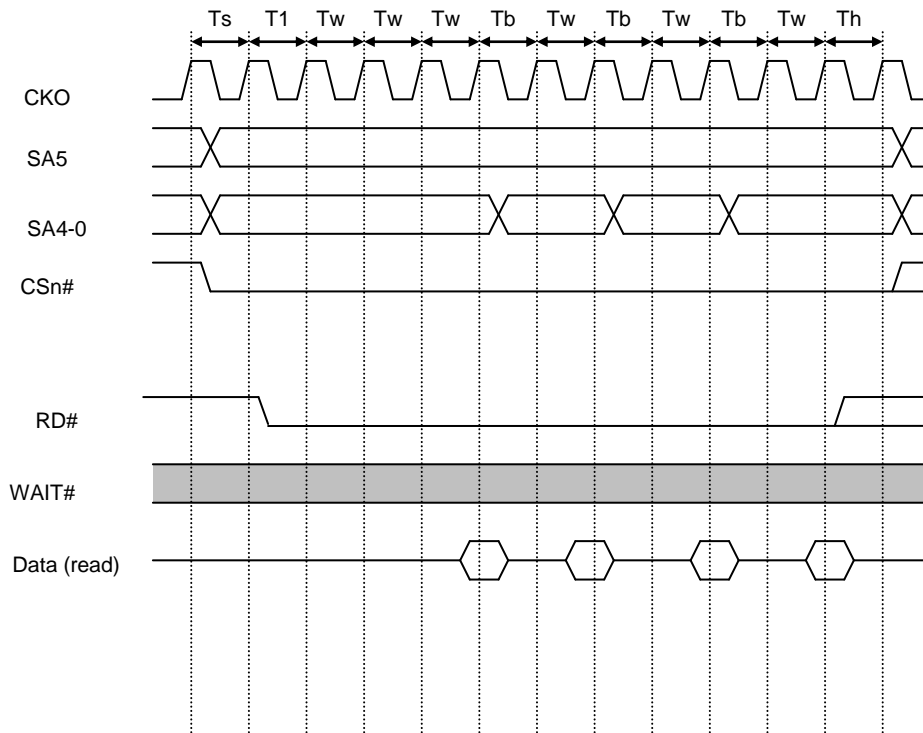
#### 16.4.4 Burst ROM Interface

Setting SMT to 1 in SMCRn allows burst ROM to be connected to bank n (n = 1 to 6). The burst ROM interface provides high-speed access to ROM that has a nibble access function. Basically, access is performed in the same way as for normal memory, but when the first cycle ends, only the address is changed before the next access is executed. When 8-bit burst ROM is connected, the number of consecutive accesses can be set as 4, 8, 16, or 32 with bits BL1~0. When 16-bit ROM is connected, 4, 8, or 16 can be set in the same way.

For burst ROM read, TAW sets the delay time from read strobe to the first data, TBP sets the delay time from consecutive address to data. Burst ROM writes have the same timing as normal memory except TAW instead of TBP is used to set the delay time of write strobe.

WAIT# pin sampling is always performed when one or more wait states are set.

Following figures show the timing of burst ROM.



\*In this example, SMT = 1, BL = 0, TAS = 1, TAW = 3, TBP = 1, TAH = 1

**Figure 16-9 Burst ROM Read Timing (Software Wait Only)**

## 16.5 NAND Flash Interface

NAND flash can be connected to static memory bank 6~ band 1. Both 8-bit and 16-bit NAND flashes (Toggle NAND or conventional NAND) are supported. A mechanism for booting from NAND flash is also supported.

### 16.5.1 Register Description

**Table 16-5 NAND Flash Interface Registers**

| Name   | Description                        | RW | Reset Value | Address    | Access Width |
|--------|------------------------------------|----|-------------|------------|--------------|
| NFCSR  | NAND flash control/status register | RW | 0x00000000  | 0x13410050 | 32           |
| PNCR   | NAND PN control register           | RW | 0x00000000  | 0x13410100 | 32           |
| PNDR   | NAND PN data register              | RW | 0x00005AA5  | 0x13410104 | 32           |
| BITCNT | NAND bit counter                   | R  | 0x00000000  | 0x13410108 | 32           |
| TGWE   | Toggle NAND Data Write Access      | RW | 0x00010000  | 0x1341010C | 32           |
| TGCR1  | Toggle NAND Control Register 1     | RW | 0x0FFF770A  | 0x13410110 | 32           |
| TGCR2  | Toggle NAND Control Register 2     | RW | 0x0FFF770A  | 0x13410114 | 32           |
| TGCR3  | Toggle NAND Control Register 3     | RW | 0x0FFF770A  | 0x13410118 | 32           |
| TGCR4  | Toggle NAND Control Register 4     | RW | 0x0FFF770A  | 0x1341011C | 32           |
| TGCR5  | Toggle NAND Control Register 5     | RW | 0x0FFF770A  | 0x13410120 | 32           |

|       |                                                               |    |            |            |    |
|-------|---------------------------------------------------------------|----|------------|------------|----|
| TGCR6 | Toggle NAND Control Register 6                                | RW | 0x0FFF770A | 0x13410124 | 32 |
| TGSR  | Toggle NAND RD# to DQS and DQ Delay Register                  | RW | 0x14A5294A | 0x13410128 | 32 |
| TGFL  | Toggle NAND ALE Fall to DQS Rise (bank 1/2/3)                 | RW | 0x00FFFFFF | 0x1341012C | 32 |
| TGFH  | Toggle NAND ALE Fall to DQS Rise (bank 4/5/6)                 | RW | 0x00FFFFFF | 0x13410130 | 32 |
| TGCL  | Toggle NAND CLE to RD# Low (bank 1/2/3)                       | RW | 0x00646464 | 0x13410134 | 32 |
| TGCH  | Toggle NAND CLE to RD# Low (bank 4/5/6)                       | RW | 0x00646464 | 0x13410138 | 32 |
| TGPD  | Toggle NAND Data Postamble Hold Time Done                     | R  | 0x0000003F | 0x1341013C | 32 |
| TGSL  | Toggle NAND DQS Setup Time for Data Input Start (bank 1/2/3)  | RW | 0x00252525 | 0x13410140 | 32 |
| TGSH  | Toggle NAND DQS Setup Time for Data Input Start (bank 4/5/6)  | RW | 0x00252525 | 0x13410144 | 32 |
| TGRR  | Toggle NAND Timer for Random Data Input and Register Read Out | RW | 0x00010000 | 0x13410148 | 32 |
| TGDR  | Toggle NAND DQS Delay Control Register                        | RW | 0x03000016 | 0x1341014C | 32 |
| TGHL  | Toggle NAND DQS Hold Time for Data Input Finish(bank 1/2/3)   | RW | 0x01252525 | 0x13410150 | 32 |
| TGHH  | Toggle NAND DQS Hold Time for Data Input Finish(bank 4/5/6)   | RW | 0x00252525 | 0x13410154 | 32 |

### 16.5.1.1 NAND Flash Control/Status Register (NFCSR)

NFCSR is a 32-bit read/write register that is used to configure NAND flash. It is initialized by any reset.

| NFCSR |      |          |    |    |    |    |    |    |    |    |    |       |       |       |       | 0x13410050 |       |          |    |    |    |    |   |       |       |       |       |       |       |      |   |   |
|-------|------|----------|----|----|----|----|----|----|----|----|----|-------|-------|-------|-------|------------|-------|----------|----|----|----|----|---|-------|-------|-------|-------|-------|-------|------|---|---|
| Bit   | 31   | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20    | 19    | 18    | 17    | 16         | 15    | 14       | 13 | 12 | 11 | 10 | 9 | 8     | 7     | 6     | 5     | 4     | 3     | 2    | 1 | 0 |
|       | DAEC | Reserved |    |    |    |    |    |    |    |    |    | TNFE6 | TNFE5 | TNFE4 | TNFE3 | TNFE2      | TNFE1 | Reserved |    |    |    |    |   | NFCE6 | NFCE5 | NFCE4 | NFCE3 | NFCE2 | NFCE1 | NFE1 |   |   |
| RST   | 0    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0     | 0     | 0          | 0     | 0        | 0  | 0  | 0  | 0  | 0 | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0 | 0 |

| Bits        | Name                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RW          |                    |   |                                                  |   |                                            |    |
|-------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------------------|---|--------------------------------------------------|---|--------------------------------------------|----|
| 31          | DAEC                                             | <b>Toggle NAND Data Access Enabel Clear:</b> write 1 to clear TGWE.DAE.                                                                                                                                                                                                                                                                                                                                                                                                        | W1          |                    |   |                                                  |   |                                            |    |
| 30:22       | Reserved                                         | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                           | R           |                    |   |                                                  |   |                                            |    |
| 21:16       | TNFE <sub>n</sub><br>(n=1,2,3,4,5,6)             | <b>Toggle NAND Flash Enable:</b> Specifies wither Toggle NAND flash or conventional NAND flash is connected to static bank n. When system is configured to boot from Toggle NAND flash, this bit is initialized to 1.<br><br><table border="0"> <tr> <td><b>TNFE</b></td> <td><b>Description</b></td> </tr> <tr> <td>0</td> <td>Static bank n is used as conventional NAND flash</td> </tr> <tr> <td>1</td> <td>Static bank n is used as Toggle NAND flash</td> </tr> </table> | <b>TNFE</b> | <b>Description</b> | 0 | Static bank n is used as conventional NAND flash | 1 | Static bank n is used as Toggle NAND flash | RW |
| <b>TNFE</b> | <b>Description</b>                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |                    |   |                                                  |   |                                            |    |
| 0           | Static bank n is used as conventional NAND flash |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |                    |   |                                                  |   |                                            |    |
| 1           | Static bank n is used as Toggle NAND flash       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |                    |   |                                                  |   |                                            |    |

| 15:12                | Reserved                                                      | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | R   |             |   |                                                               |   |                                     |    |
|----------------------|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------------|---|---------------------------------------------------------------|---|-------------------------------------|----|
| 1/3/5/<br>7/9/1<br>1 | FCEn<br>(n=1,2,3,<br>4,5,6)                                   | <p><b>NAND Flash FCE# Assertion Control:</b> Controls the assertion of NAND Flash FCEn#. When set, FCEn# is always asserted until this bit is cleared. When the NAND flash (Toggle NAND flash or conventional NAND flash) require FCEn# to be asserted during read busy time, this bit should be set.</p> <p>NOTE:<br/>Before set FCEn to 1, DPHTDn MUST be checked by software.<br/>When the last Toggle NAND data write or read access done, software MUST clear the relevant FCEn.</p> <table border="1"> <thead> <tr> <th>FCE</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>FCEn# is asserted as normal static chip enable(Initial value)</td> </tr> <tr> <td>1</td> <td>FCEn# is always asserted</td> </tr> </tbody> </table> | FCE | Description | 0 | FCEn# is asserted as normal static chip enable(Initial value) | 1 | FCEn# is always asserted            | RW |
| FCE                  | Description                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |             |   |                                                               |   |                                     |    |
| 0                    | FCEn# is asserted as normal static chip enable(Initial value) |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |             |   |                                                               |   |                                     |    |
| 1                    | FCEn# is always asserted                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |             |   |                                                               |   |                                     |    |
| 0/2/4/<br>6/8/1<br>0 | NFEn<br>(n=1,2,3,<br>4,5,6)                                   | <p><b>NAND Flash Enable:</b> Specifies if NAND flash (Toggle NAND flash or conventional NAND flash) is connected to static bank n. When system is configured to boot from NAND flash, this bit is initialized to 1.</p> <table border="1"> <thead> <tr> <th>NFE</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Static bank n is not used as NAND flash</td> </tr> <tr> <td>1</td> <td>Static bank n is used as NAND flash</td> </tr> </tbody> </table>                                                                                                                                                                                                                                                                              | NFE | Description | 0 | Static bank n is not used as NAND flash                       | 1 | Static bank n is used as NAND flash | RW |
| NFE                  | Description                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |             |   |                                                               |   |                                     |    |
| 0                    | Static bank n is not used as NAND flash                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |             |   |                                                               |   |                                     |    |
| 1                    | Static bank n is used as NAND flash                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |             |   |                                                               |   |                                     |    |

### NAND PN Control Register (PNCR)

PNCR is a 32-bit read/write register that is used to control NAND flash data randomization. It is initialized by any reset.

|             |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |         |         |        |          |       |      |   |   |   |   |   |   |   |   |   |   |
|-------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|---------|---------|--------|----------|-------|------|---|---|---|---|---|---|---|---|---|---|
| <b>PNCR</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13410100</b> |         |         |        |          |       |      |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15      | 14      | 13     | 12       | 11    | 10   | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   | BIT_RST | BIT_SEL | BIT_EN | Reserved | PNRST | PNEN |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0       | 0       | 0      | 0        | 0     | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits    | Name                                      | Description                                                                                                                                                                                                                                                                                                                            | RW      |             |   |                                           |   |                                           |    |
|---------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------------|---|-------------------------------------------|---|-------------------------------------------|----|
| 31:6    | Reserved                                  | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                   | R       |             |   |                                           |   |                                           |    |
| 5       | BIT_RST                                   | <b>NAND BIT Counter Reset:</b> Reset Bit counter. When this bit is written to 1, the bit counter is reset to 0. This bit is write-only.                                                                                                                                                                                                | W       |             |   |                                           |   |                                           |    |
| 4       | BIT_SEL                                   | <p><b>NAND BIT Counter Select:</b> Bit counter’s counting select.</p> <table border="1"> <thead> <tr> <th>BIT_SEL</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Calculate number of “0” in NAND read data</td> </tr> <tr> <td>1</td> <td>Calculate number of “1” in NAND read Data</td> </tr> </tbody> </table> | BIT_SEL | Description | 0 | Calculate number of “0” in NAND read data | 1 | Calculate number of “1” in NAND read Data | RW |
| BIT_SEL | Description                               |                                                                                                                                                                                                                                                                                                                                        |         |             |   |                                           |   |                                           |    |
| 0       | Calculate number of “0” in NAND read data |                                                                                                                                                                                                                                                                                                                                        |         |             |   |                                           |   |                                           |    |
| 1       | Calculate number of “1” in NAND read Data |                                                                                                                                                                                                                                                                                                                                        |         |             |   |                                           |   |                                           |    |
| 3       | BIT_EN                                    | <b>NAND BIT Counter Enable:</b> Enable/disable bit counter counting.                                                                                                                                                                                                                                                                   | RW      |             |   |                                           |   |                                           |    |

|   |          | <b>BIT_EN Description</b>                                                                                                                                                                                                          |    |
|---|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|   |          | 0 Bit counting is disabled<br>1 Bit counting is enabled                                                                                                                                                                            |    |
| 2 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                               | R  |
| 1 | PNRST    | <b>NAND Flash PN Reset:</b> Reset seed of randomizer. When this bit is written to 1, the seed of randomizer is reset. This bit is write-only.                                                                                      | W  |
| 0 | PNEN     | <b>NAND Flash PN Enable:</b> Specifies if NAND flash read/write data randomization is enabled. This bit is initialized to 0.<br><br><b>PNEN Description</b><br>0 Data randomization is disabled<br>1 Data randomization is enabled | RW |

### 16.5.1.2 NAND PN Data Register (PNDR)

PNDR is a 23-bit read/write register that is used for seed of randomizer during NAND read/write.

| <b>PNDR</b> |          |    |    |    |    |    |    |    |    |      |    |    |    |    |    | <b>0x13410104</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------------|----------|----|----|----|----|----|----|----|----|------|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit         | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22   | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             | Reserved |    |    |    |    |    |    |    |    | PNDR |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST         | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0                 | 0  | 1  | 0  | 1  | 1  | 0  | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

### 16.5.1.3 NAND Bit Counter (BITCNT)

BITCNT is a 32-bit read/write register that is used to counting the number of “1” or “0” (based on BIT\_SEL) in Nand read data and keep counting during Nand read till BIT Counter Reset. It is initialized by any reset.

| <b>BITCNT</b> |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13410108</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|---------------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| Bit           | 31     | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|               | BITCNT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST           | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 16.5.1.4 Toggle NAND Data Write Access (TGWE)

DQS Setup Time for Data Write Start (TGSL.CDQSS and TGSH.CDQSS) is defined from the last data input condition of the control signals such as CE, CLE and ALE. Before any Toggle NAND data write access, software MUST write 1 to corresponding SDEn, and WCD shall be checked prior to a write data access.

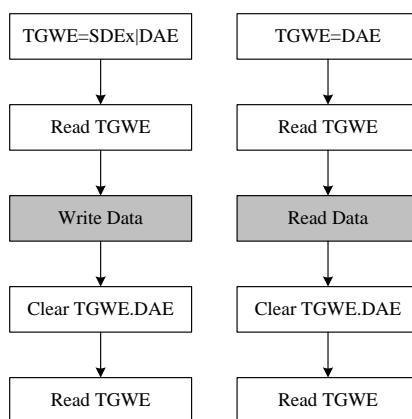


Figure 16-10 Toggle NAND Page Write/Read Operation

| TGWE |     |       |          |    |    |    |    |    |    |    |    |    |    |    |    | 0x1341010C |          |    |    |    |    |    |      |      |      |      |      |      |   |   |   |   |
|------|-----|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|------------|----------|----|----|----|----|----|------|------|------|------|------|------|---|---|---|---|
| Bit  | 31  | 30    | 29       | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15       | 14 | 13 | 12 | 11 | 10 | 9    | 8    | 7    | 6    | 5    | 4    | 3 | 2 | 1 | 0 |
|      | DAE | FIDLE | Reserved |    |    |    |    |    |    |    |    |    |    |    |    | WCD        | Reserved |    |    |    |    |    | SDE6 | SDE5 | SDE4 | SDE3 | SDE2 | SDE1 |   |   |   |   |
| RST  | 0   | 1     | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1          | 0        | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0    | 0    | 0    | 0    | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                              | RW |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | DAE      | <b>Toggle NAND Data Access Enabel:</b> Before Data Access (Data Write & Data Read) this bit MUST set to 1, and after Data Access DAE MUST clear by software (before next Command or Address cycle).<br><br><b>NOTE.</b> DAE can be cleared by set <b>NFCSR.DAEC=1</b> .                  | WR |
| 30    | FIDLE    | <b>Internal FSM IDLE</b>                                                                                                                                                                                                                                                                 | R  |
| 29:17 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                     | R  |
| 16    | WCD      | <b>DQS Setup Time for data input start Done:</b> After set SDEn, this bit will be low CDQSS cycles. When WCD return to 1 data can be write to Toggle NAND.<br><br><b>WCD Description</b><br>0 DQS Setup Time for Data Write Start not pass<br>1 DQS Setup Time for Data Input Start Pass | R  |
| 15:6  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                     | R  |
| 5     | SDE6     | <b>Set DQS output enable (bank6):</b> Before data write access, software MUST write 1 to SDE6. After CDQSS cycles, controller will goes into data-write-mode.                                                                                                                            | W  |
| 4     | SDE5     | <b>Set DQS output enable (bank5):</b> Before data write access, software MUST write 1 to SDE5. After CDQSS cycles, controller will goes into data-write-mode.                                                                                                                            | W  |



|   |      |                                                                                                                                                               |   |
|---|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 3 | SDE4 | <b>Set DQS output enable (bank4):</b> Before data write access, software MUST write 1 to SDE4. After CDQSS cycles, controller will goes into data-write-mode. | W |
| 2 | SDE3 | <b>Set DQS output enable (bank3):</b> Before data write access, software MUST write 1 to SDE3. After CDQSS cycles, controller will goes into data-write-mode. | W |
| 1 | SDE2 | <b>Set DQS output enable (bank2):</b> Before data write access, software MUST write 1 to SDE2. After CDQSS cycles, controller will goes into data-write-mode. | W |
| 0 | SDE1 | <b>Set DQS output enable (bank1):</b> Before data write access, software MUST write 1 to SDE1. After CDQSS cycles, controller will goes into data-write-mode. | W |

### 16.5.1.5 Toggle NAND Control Register (TGCR1~6)

TGCR1~6 are 32-bit read/write registers that contain control bits for Toggle NAND flash. On reset, TGCR1~6 are initialized to 0x0FFF770A.

NOTE:

- (1) If the conventional type NAND (i.e. SDR NAND) enable, all of the Command Latch Cycle/Address Latch Cycle/Data input Timing/Data output Timing/Read ID Operation/Read Status Cycle will be configured by SMCRn. And Toggle NAND Control Register (TGCRn) will be ignored.
- (2) However, if the high-speed Toggle DDR NAND enable, only the Command Latch Cycle/Address Latch Cycle are configured by SMCRn, but the Data input Timing/Data output Timing/Read ID Operation/Read Status Cycle are configured by Toggle NAND Control Register(TGCRn).

|       |          |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |          |   |   |   |      |   |   |   |   |   |            |
|-------|----------|-----|----|----|----|-----|----|----|----|-----|----|----|----|-----|----|----|----|-----|----|----|----|----------|---|---|---|------|---|---|---|---|---|------------|
| TGCR1 |          |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |          |   |   |   |      |   |   |   |   |   | 0x13410110 |
| TGCR2 |          |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |          |   |   |   |      |   |   |   |   |   | 0x13410114 |
| TGCR3 |          |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |          |   |   |   |      |   |   |   |   |   | 0x13410118 |
| TGCR4 |          |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |          |   |   |   |      |   |   |   |   |   | 0x1341011C |
| TGCR5 |          |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |          |   |   |   |      |   |   |   |   |   | 0x13410120 |
| TGCR6 |          |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |     |    |    |    |          |   |   |   |      |   |   |   |   |   | 0x13410124 |
| Bit   | 31       | 30  | 29 | 28 | 27 | 26  | 25 | 24 | 23 | 22  | 21 | 20 | 19 | 18  | 17 | 16 | 15 | 14  | 13 | 12 | 11 | 10       | 9 | 8 | 7 | 6    | 5 | 4 | 3 | 2 | 1 | 0          |
|       | Reserved | TRV |    |    |    | TRW |    |    |    | TWW |    |    |    | TAH |    |    |    | TAS |    |    |    | Reserved |   |   |   | DPHT |   |   |   |   |   |            |
| RST   | 0        | 0   | 0  | 0  | 1  | 1   | 1  | 1  | 1  | 1   | 1  | 1  | 1  | 1   | 1  | 1  | 0  | 1   | 1  | 1  | 0  | 1        | 1 | 1 | 0 | 0    | 0 | 0 | 1 | 0 | 1 | 0          |

| Bits  | Name     | Description                                                                                                                                                                                | RW |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | Reserved | Writing has no effect, read as zero.                                                                                                                                                       | R  |
| 29:24 | TRV      | <b>Toggle NAND Flash Recovery Time.</b><br>Its value is the number of idle cycles (0~63 cycles) inserted between bus cycles when switching from one bank to another bank or between a read | RW |

|       |            | access to a write access in the same bank. Its initial value is 0xF (15 cycles).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
|-------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------------|-----------|------|----------|---------|------|----------|---------|------|----------|---------|------|----------|---------|------|----------|---------|------|----------|---------|------|----------|---------|------|----------|---------|------|-----------|------------------------|------|-----------|---------|------|-----------|---------|------|-----------|---------|------|-----------|---------|------|-----------|---------|----|
| 23:20 | TRW        | <b>Toggle NAND Flash Read Wait Time.</b><br>These bits specify the number of wait cycles to be inserted in read strobe time.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW   |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 19:16 | TWW        | <b>Toggle NAND Flash Write Wait Time.</b><br>These bits specify the number of wait cycles to be inserted in write strobe time.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW   |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 15:12 | TAH        | <b>Toggle NAND Flash Address Hold Time.</b><br>These bits specify the number of wait cycles to be inserted from negation of read/write strobe to address. AHB cycles=TAH+1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW   |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 11:8  | TAS        | <b>Toggle NAND Flash Address Setup Time.</b><br>These bits specify the number of wait cycles (1~15 cycles) to be inserted from address to assertion of read/write strobe.<br><b>NOTE. TAS can NOT be 0. AHB cycles=TAS+1</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW   |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 7:4   | Reserved   | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | R    |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 3:0   | DPHT       | <b>Toggle NAND Flash Data Postamble Hold Time.</b><br>CE# is once deasserted, it shall stay high over at least DPHT cycles before it is asserted again. Therefore, software MUST wait corresponding DPHTDn return to 1 before set FCEn.<br><br><table border="1" style="margin-left: 40px;"> <thead> <tr> <th>DPHT</th> <th>Wait cycle</th> <th>Wait# Pin</th> </tr> </thead> <tbody> <tr><td>0010</td><td>2 cycles</td><td>Enabled</td></tr> <tr><td>0011</td><td>3 cycles</td><td>Enabled</td></tr> <tr><td>0100</td><td>4 cycles</td><td>Enabled</td></tr> <tr><td>0101</td><td>5 cycles</td><td>Enabled</td></tr> <tr><td>0110</td><td>6 cycles</td><td>Enabled</td></tr> <tr><td>0111</td><td>7 cycles</td><td>Enabled</td></tr> <tr><td>1000</td><td>8 cycles</td><td>Enabled</td></tr> <tr><td>1001</td><td>9 cycles</td><td>Enabled</td></tr> <tr><td>1010</td><td>10 cycles</td><td>Enabled(Initial Value)</td></tr> <tr><td>1011</td><td>12 cycles</td><td>Enabled</td></tr> <tr><td>1100</td><td>15 cycles</td><td>Enabled</td></tr> <tr><td>1101</td><td>20 cycles</td><td>Enabled</td></tr> <tr><td>1110</td><td>25 cycles</td><td>Enabled</td></tr> <tr><td>1111</td><td>31 cycles</td><td>Enabled</td></tr> </tbody> </table> <b>NOTE: DPHT MUST &gt;= 2.</b> | DPHT | Wait cycle | Wait# Pin | 0010 | 2 cycles | Enabled | 0011 | 3 cycles | Enabled | 0100 | 4 cycles | Enabled | 0101 | 5 cycles | Enabled | 0110 | 6 cycles | Enabled | 0111 | 7 cycles | Enabled | 1000 | 8 cycles | Enabled | 1001 | 9 cycles | Enabled | 1010 | 10 cycles | Enabled(Initial Value) | 1011 | 12 cycles | Enabled | 1100 | 15 cycles | Enabled | 1101 | 20 cycles | Enabled | 1110 | 25 cycles | Enabled | 1111 | 31 cycles | Enabled | RW |
| DPHT  | Wait cycle | Wait# Pin                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 0010  | 2 cycles   | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 0011  | 3 cycles   | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 0100  | 4 cycles   | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 0101  | 5 cycles   | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 0110  | 6 cycles   | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 0111  | 7 cycles   | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 1000  | 8 cycles   | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 1001  | 9 cycles   | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 1010  | 10 cycles  | Enabled(Initial Value)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 1011  | 12 cycles  | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 1100  | 15 cycles  | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 1101  | 20 cycles  | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 1110  | 25 cycles  | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |
| 1111  | 31 cycles  | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |            |           |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |          |         |      |           |                        |      |           |         |      |           |         |      |           |         |      |           |         |      |           |         |    |

### 16.5.1.6 Toggle NAND RD# to DQS and DQ delay Register (TGSR)

TGSR is 32-bit read/write registers that specify the number of delay cycles (0~31 cycles) from RD# to DQS for Toggle NAND flash. On reset, TGSR is initialized to 0x14A5294A. See Figure 16-12.

| TGSR |          |    |        |    |    |    |        |    |    |    |        |    |    |    |        | 0x13410128 |    |    |        |    |    |    |        |   |   |   |   |   |   |   |   |   |
|------|----------|----|--------|----|----|----|--------|----|----|----|--------|----|----|----|--------|------------|----|----|--------|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| Bit  | 31       | 30 | 29     | 28 | 27 | 26 | 25     | 24 | 23 | 22 | 21     | 20 | 19 | 18 | 17     | 16         | 15 | 14 | 13     | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|      | Reserved |    | DQSRE5 |    |    |    | DQSRE5 |    |    |    | DQSRE4 |    |    |    | DQSRE3 |            |    |    | DQSRE2 |    |    |    | DQSRE1 |   |   |   |   |   |   |   |   |   |
| RST  | 0        | 0  | 0      | 1  | 0  | 1  | 0      | 0  | 1  | 0  | 1      | 0  | 0  | 1  | 0      | 1          | 0  | 0  | 1      | 0  | 0  | 1  | 0      | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

| Bits  | Name     | Description                                                                                  | RW |
|-------|----------|----------------------------------------------------------------------------------------------|----|
| 31:30 | Reserved | Writing has no effect, read as zero.                                                         | R  |
| 29:25 | DQSRE6   | <b>Toggle NAND Flash RD# to DQS and DQ delay (Bank6).</b><br>10 cycles is the initial value. | RW |
| 24:20 | DQSRE5   | <b>Toggle NAND Flash RD# to DQS and DQ delay (Bank5).</b><br>10 cycles is the initial value. | RW |
| 19:15 | DQSRE4   | <b>Toggle NAND Flash RD# to DQS and DQ delay (Bank4).</b><br>10 cycles is the initial value. | RW |
| 14:10 | DQSRE3   | <b>Toggle NAND Flash RD# to DQS and DQ delay (Bank3).</b><br>10 cycles is the initial value. | RW |
| 9:5   | DQSRE2   | <b>Toggle NAND Flash RD# to DQS and DQ delay (Bank2).</b><br>10 cycles is the initial value. | RW |
| 4:0   | DQSRE1   | <b>Toggle NAND Flash RD# to DQS and DQ delay (Bank1).</b><br>10 cycles is the initial value. | RW |

### 16.5.1.7 Toggle NAND ALE Fall to DQS Rise (bank 1/2/3 TGFL)

TGFL is 32-bit read/write registers that specify the number of wait cycles (0~255 cycles) from ALE fall to DQS rise for bank 1/2/3. On reset, TGFL is initialized to 0x00FFFFFF.

| TGFL |          |    |    |    |    |    |    |    |      |    |    |    |    |    |    | 0x1341012C |      |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|------|----------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|------------|------|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
| Bit  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23   | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|      | Reserved |    |    |    |    |    |    |    | FDA3 |    |    |    |    |    |    |            | FDA2 |    |    |    |    |    |   |   | FDA1 |   |   |   |   |   |   |   |
| RST  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1    | 1  | 1  | 1  | 1  | 1  | 1  | 1          | 1    | 1  | 1  | 1  | 1  | 1  | 1 | 1 | 1    | 1 | 1 | 1 | 1 | 1 | 1 |   |

| Bits  | Name     | Description                                                                                | RW |
|-------|----------|--------------------------------------------------------------------------------------------|----|
| 31:24 | Reserved | Writing has no effect, read as zero.                                                       | R  |
| 23:16 | FDA 3    | <b>Toggle NAND Flash ALE Fall to DQS Rise (Bank3).</b><br>255 cycles is the initial value. | RW |
| 15:8  | FDA 2    | <b>Toggle NAND Flash ALE Fall to DQS Rise (Bank2).</b><br>255 cycles is the initial value. | RW |
| 7:0   | FDA 1    | <b>Toggle NAND Flash ALE Fall to DQS Rise (Bank1).</b><br>255 cycles is the initial value. | RW |

### 16.5.1.8 Toggle NAND ALE Fall to DQS Rise (bank 4/5/6 TGFH)

TGFH is 32-bit read/write registers that specify the number of wait cycles (0~255 cycles) from ALE fall to DQS rise for bank 4/5/6. On reset, TGFH is initialized to 0x00FFFFFF.

| TGFH |          |    |    |    |    |    |    |    |      |    |    |    |      |    |    | 0x13410134 |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |
|------|----------|----|----|----|----|----|----|----|------|----|----|----|------|----|----|------------|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|
| Bit  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23   | 22 | 21 | 20 | 19   | 18 | 17 | 16         | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |  |  |  |  |
|      | Reserved |    |    |    |    |    |    |    | FDA6 |    |    |    | FDA5 |    |    |            | FDA4 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |
| RST  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1    | 1  | 1  | 1  | 1    | 1  | 1  | 1          | 1    | 1  | 1  | 1  | 1  | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |  |

| Bits  | Name     | Description                                                                                      | RW |
|-------|----------|--------------------------------------------------------------------------------------------------|----|
| 31:24 | Reserved | Writing has no effect, read as zero.                                                             | R  |
| 23:16 | FDA 6    | <b>Toggle NAND Flash First ALE Fall to DQS Rise (Bank6).</b><br>255 cycles is the initial value. | RW |
| 15:8  | FDA 5    | <b>Toggle NAND Flash First ALE Fall to DQS Rise (Bank5).</b><br>255 cycles is the initial value. | RW |
| 7:0   | FDA 4    | <b>Toggle NAND Flash First ALE Fall to DQS Rise (Bank4).</b><br>255 cycles is the initial value. | RW |

### 16.5.1.9 Toggle NAND CLE to RD# Low (bank 1/2/3 TGCL)

TGCL is 32-bit read/write registers that specify the number of wait cycles (0~255 cycles) from CLE to RD# low for bank 1/2/3. On reset, TGCL is initialized to 0x00646464.

| TGCL |          |    |    |    |    |    |    |    |      |    |    |    |      |    |    | 0x13410134 |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|------|----------|----|----|----|----|----|----|----|------|----|----|----|------|----|----|------------|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| Bit  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23   | 22 | 21 | 20 | 19   | 18 | 17 | 16         | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
|      | Reserved |    |    |    |    |    |    |    | CLR3 |    |    |    | CLR2 |    |    |            | CLR1 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| RST  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 1  | 1  | 0  | 0    | 1  | 0  | 0          | 0    | 1  | 1  | 0  | 0  | 1  | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |  |  |  |  |  |  |

| Bits  | Name     | Description                                                                                                                          | RW |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:24 | Reserved | Writing has no effect, read as zero.                                                                                                 | R  |
| 23:16 | CLR 3    | <b>Toggle NAND Flash CLE to RE_n Low (Bank3).</b><br>100 cycles is the initial value.<br>NOTE. CLR MUST smaller than FDA3 , and >=2. | RW |
| 15:8  | CLR 2    | <b>Toggle NAND Flash CLE to RE_n Low (Bank2).</b><br>100 cycles is the initial value.                                                | RW |

|     |       |                                                                                                                                           |    |
|-----|-------|-------------------------------------------------------------------------------------------------------------------------------------------|----|
|     |       | NOTE. CLR MUST smaller than FDA2, and $\geq 2$ .                                                                                          |    |
| 7:0 | CLR 1 | <b>Toggle NAND Flash CLE to RE_n Low (Bank1).</b><br>100 cycles is the initial value.<br>NOTE. CLR MUST smaller than FDA1, and $\geq 2$ . | RW |

### 16.5.1.10 Toggle NAND CLE to RD# low (bank 4/5/6 TGCH)

TGCH is 32-bit read/write registers that specify the number of wait cycles (0~255 cycles) from CLE to RD# low for bank 4/5/6. On reset, TGCH is initialized to 0x00646464.

|            |             |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |                   |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
|            | <b>TGCH</b> |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    | <b>0x13410138</b> |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23   | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|            | Reserved    |    |    |    |    |    |    |    | CLR6 |    |    |    |    |    |    |    | CLR5              |    |    |    |    |    |   |   | CLR4 |   |   |   |   |   |   |   |
| <b>RST</b> | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0                 | 1  | 1  | 0  | 0  | 1  | 0 | 0 | 0    | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                               | RW |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:24 | Reserved | Writing has no effect, read as zero.                                                                                                      | R  |
| 23:16 | CLR 6    | <b>Toggle NAND Flash CLE to RE_n Low (Bank3).</b><br>100 cycles is the initial value.<br>NOTE. CLR MUST smaller than FDA6, and $\geq 2$ . | RW |
| 15:8  | CLR 5    | <b>Toggle NAND Flash CLE to RE_n Low (Bank2).</b><br>100 cycles is the initial value.<br>NOTE. CLR MUST smaller than FDA5, and $\geq 2$ . | RW |
| 7:0   | CLR 4    | <b>Toggle NAND Flash CLE to RE_n Low (Bank1).</b><br>100 cycles is the initial value.<br>NOTE. CLR MUST smaller than FDA4, and $\geq 2$ . | RW |

### 16.5.1.11 Toggle NAND Data Postamble Hold Time Done (TGPD)

If CE# is once deasserted, it shall stay high over at least DPHT cycles before it is asserted again. Therefore, DPHTDn shall be checked prior to any chip re-enable (i.e. set FCEn).

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |   |   |        |        |        |        |        |        |   |   |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|---|---|
|            | <b>TGPD</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x1341013C</b> |    |    |    |    |    |   |   |        |        |        |        |        |        |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1 | 0 |
|            | Reserved    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |   |   | DPHTD6 | DPHTD5 | DPHTD4 | DPHTD3 | DPHTD2 | DPHTD1 |   |   |
| <b>RST</b> | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0      | 0      | 1      | 1      | 1      | 1      | 1 | 1 |

| Bits | Name     | Description                                                                                                                                                                    | RW |
|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:6 | Reserved | Writing has no effect, read as zero.                                                                                                                                           | R  |
| 5    | DPHTD6   | <b>Toggle NAND Flash Data Postamble Hold Time Done (Bank6).</b><br><b>DPHTD6 Description</b><br>0 Can NOT enable CE6#<br>1 Data Postamble Hold Time Done, FCE6 can be set to 1 | R  |
| 4    | DPHTD5   | <b>Toggle NAND Flash Data Postamble Hold Time Done (Bank5).</b><br><b>DPHTD5 Description</b><br>0 Can NOT enable CE5#<br>1 Data Postamble Hold Time Done, FCE5 can be set to 1 | R  |
| 3    | DPHTD4   | <b>Toggle NAND Flash Data Postamble Hold Time Done (Bank4).</b><br><b>DPHTD4 Description</b><br>0 Can NOT enable CE4#<br>1 Data Postamble Hold Time Done, FCE4 can be set to 1 | R  |
| 2    | DPHTD3   | <b>Toggle NAND Flash Data Postamble Hold Time Done (Bank3).</b><br><b>DPHTD3 Description</b><br>0 Can NOT enable CE3#<br>1 Data Postamble Hold Time Done, FCE3 can be set to 1 | R  |
| 1    | DPHTD2   | <b>Toggle NAND Flash Data Postamble Hold Time Done (Bank2).</b><br><b>DPHTD2 Description</b><br>0 Can NOT enable CE2#<br>1 Data Postamble Hold Time Done, FCE2 can be set to 1 | R  |
| 0    | DPHTD1   | <b>Toggle NAND Flash Data Postamble Hold Time Done (Bank1).</b><br><b>DPHTD1 Description</b><br>0 Can NOT enable CE1#<br>1 Data Postamble Hold Time Done, FCE1 can be set to 1 | R  |

### 16.5.1.12 Toggle NAND DQS Setup Time for Data Input Start (bank 1/2/3 TGSL)

DQS Setup Time for Data Write Start (i.e. TGSL.CDQSS) is defined from the last data input condition of the control signals such as CE, CLE and ALE.

|             |          |    |    |    |    |    |    |    |         |    |    |    |    |    |    |                   |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
|-------------|----------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|-------------------|---------|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
| <b>TGSL</b> |          |    |    |    |    |    |    |    |         |    |    |    |    |    |    | <b>0x13410140</b> |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
| <b>Bit</b>  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             | Reserved |    |    |    |    |    |    |    | CDQSS 3 |    |    |    |    |    |    |                   | CDQSS 2 |    |    |    |    |    |   |   | CDQSS 1 |   |   |   |   |   |   |   |
| <b>RST</b>  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 1  | 0  | 0  | 1  | 0  | 1                 | 0       | 0  | 1  | 0  | 0  | 1  | 0 | 1 | 0       | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

| Bits  | Name       | Description                                                                                                                   | RW |
|-------|------------|-------------------------------------------------------------------------------------------------------------------------------|----|
| 31:24 | Reserved   | Writing has no effect, read as zero.                                                                                          | R  |
| 23:16 | CDQSS<br>3 | <b>DQS Setup Time for data input start (bank3).</b><br>(i.e. ale_fall – dqs_rise < tCDQSS3 and ale_fall – dqs_fall < tCDQSS3) | RW |

|      |            |                                                                                                                               |    |
|------|------------|-------------------------------------------------------------------------------------------------------------------------------|----|
| 15:8 | CDQSS<br>2 | <b>DQS Setup Time for data input start (bank2).</b><br>(i.e. ale_fall – dqs_rise < tCDQSS2 and ale_fall – dqs_fall < tCDQSS2) | RW |
| 7:0  | CDQSS<br>1 | <b>DQS Setup Time for data input start (bank1).</b><br>(i.e. ale_fall – dqs_rise < tCDQSS1 and ale_fall – dqs_fall < tCDQSS1) | RW |

### 16.5.1.13 Toggle NAND DQS Setup Time for Data Input Start (bank 4/5/6 TGSH)

DQS Setup Time for Data Write Start (TGSH.CDQSS) is defined from the last data input condition of the control signals such as CE, CLE and ALE.

|            |             |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |                   |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
|            | <b>TGSH</b> |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    | <b>0x13410144</b> |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|            | Reserved    |    |    |    |    |    |    |    | CDQSS 6 |    |    |    |    |    |    |    | CDQSS 5           |    |    |    |    |    |   |   | CDQSS 4 |   |   |   |   |   |   |   |
| <b>RST</b> | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 1  | 0  | 0  | 1  | 0  | 1  | 0                 | 0  | 1  | 0  | 0  | 1  | 0 | 1 | 0       | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

| Bits  | Name       | Description                                                                                                                   | RW |
|-------|------------|-------------------------------------------------------------------------------------------------------------------------------|----|
| 31:24 | Reserved   | Writing has no effect, read as zero.                                                                                          | R  |
| 23:16 | CDQSS<br>6 | <b>DQS Setup Time for data input start (bank6).</b><br>(i.e. ale_fall – dqs_rise < tCDQSS6 and ale_fall – dqs_fall < tCDQSS6) | RW |
| 15:8  | CDQSS<br>5 | <b>DQS Setup Time for data input start (bank5).</b><br>(i.e. ale_fall – dqs_rise < tCDQSS5 and ale_fall – dqs_fall < tCDQSS5) | RW |
| 7:0   | CDQSS<br>4 | <b>DQS Setup Time for data input start (bank4).</b><br>(i.e. ale_fall – dqs_rise < tCDQSS4 and ale_fall – dqs_fall < tCDQSS4) | RW |

### 16.5.1.14 Toggle NAND Timer for Random Data Input and Register Read Out (TGRR)

In Random Data Input or Register Read Out mode, TGRR.CWAW specifies the number of wait cycles to be inserted between Command Write Cycle and Address Write Cycle. See 1.5.4 (Program Operation with Random Data Input).

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |   |   |   |   |      |   |   |   |   |   |   |  |  |  |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|---|---|---|---|------|---|---|---|---|---|---|--|--|--|
|            | <b>TGRR</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13410148</b> |    |    |    |    |    |   |   |   |   |      |   |   |   |   |   |   |  |  |  |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5    | 4 | 3 | 2 | 1 | 0 |   |  |  |  |
|            | Reserved    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TD | Reserved          |    |    |    |    |    |   |   |   |   | CWAW |   |   |   |   |   |   |  |  |  |
| <b>RST</b> | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0                 | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0    | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |

| Bits  | Name     | Description                                                               | RW |
|-------|----------|---------------------------------------------------------------------------|----|
| 31:17 | Reserved | Writing has no effect, read as zero.                                      | R  |
| 16    | TD       | <b>Timer Done:</b> After set CWAW, this bit will be low CWAW cycles. When | R  |

|      |          |                                                                                                                                                                                                                                                                                                                                                     |   |
|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|      |          | TD return to high, address can be write to Toggle NAND.<br><b>TD Description</b><br>0 Command Write Cycle to Address Write Cycle not pass<br>1 Command Write Cycle to Address Write Cycle Pass                                                                                                                                                      |   |
| 15:8 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                | R |
| 7:0  | CWAW     | <b>Command Write Cycle to Address Write Cycle Time:</b> In Random Data Input or Register Read Out mode, these bits specify the number of wait cycles to be inserted between Command Write Cycle and Address Write Cycle.<br>Before data write access, software MUST write 1 to SDE1. After CDQSS cycles, controller will goes into data-write-mode. | W |

### 16.5.1.15 Toggle NAND DQS Delay Control Register (TGDR)

TGDR contains an on-chip DLL to control the DQS Delay for read data.

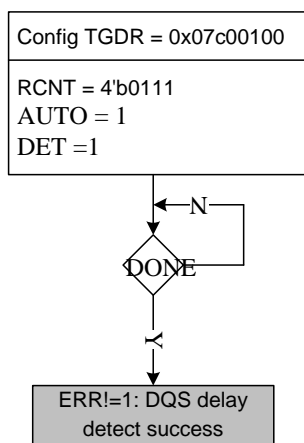


Figure 16-11 Example of DQS Delay Line Auto-Detect-Operation

| TGDR |          |     |      |      |    |    |    |     |      |          |      |    |    |    |          | 0x1341014C |    |    |    |          |      |    |   |   |   |   |   |   |   |   |   |   |
|------|----------|-----|------|------|----|----|----|-----|------|----------|------|----|----|----|----------|------------|----|----|----|----------|------|----|---|---|---|---|---|---|---|---|---|---|
| Bit  | 31       | 30  | 29   | 28   | 27 | 26 | 25 | 24  | 23   | 22       | 21   | 20 | 19 | 18 | 17       | 16         | 15 | 14 | 13 | 12       | 11   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|      | Reserved | ERR | DONE | RCNT |    |    |    | DET | AUTO | Reserved | ADQS |    |    |    | Reserved | IDQS       |    |    |    | Reserved | RDQS |    |   |   |   |   |   |   |   |   |   |   |
| RST  | 0        | 0   | 0    | 0    | 0  | 0  | 1  | 1   | 0    | 0        | 0    | 0  | 0  | 0  | 0        | 1          | 0  | 0  | 0  | 0        | 0    | 0  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |

| Bits  | Name     | Description                                                                                             | RW |
|-------|----------|---------------------------------------------------------------------------------------------------------|----|
| 31:30 | Reserved | Writing has no effect, read as zero.                                                                    | R  |
| 29    | ERR      | <b>DQS Delay Detect ERROR:</b><br>When hardware detect one ahb_clk cycle delay failed, ERR is set high; | R  |



|       |          |                                                                                                                                                                                                                                                            |    |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|       |          | ERR is cleared zero when a new detection starts;<br>ERR is valid only when DONE=1.                                                                                                                                                                         |    |
| 28    | DONE     | <b>Delay Detect Done:</b><br>When hardware detect complete, ERR is valid.<br>DONE is cleared zero when a new detection starts.                                                                                                                             | R  |
| 27:24 | RCNT     | <b>Re-Detect Wait Cycle.</b>                                                                                                                                                                                                                               | RW |
| 23    | DET      | <b>Start Delay Detecting:</b><br>Write 1 to START bit starts a new delay detect progress.                                                                                                                                                                  | W  |
| 22    | AUTO     | <b>Hardware Auto-detect &amp; Set Delay Line.</b><br>0: Hardware does NOT auto-set delay line.<br>1: Hardware auto-set delay line after detect success.                                                                                                    | RW |
| 21    | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                       | R  |
| 20:16 | ADQS     | <b>Number of Delay Elements Detect by Hardware</b>                                                                                                                                                                                                         | R  |
| 15:13 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                       | R  |
| 12:8  | IDQS     | <b>Initial Number of Delay Elements Used on Auto Detect</b>                                                                                                                                                                                                | RW |
| 7:5   | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                       | R  |
| 4:0   | RDQS     | <b>Number of Delay Elements Used on the Read DQS Delay-line:</b><br>Set the number of delay elements used on the read DQS delay-line.<br>When RDQS increase one, the delay value of read DQS increase approximately 0.3 ns.<br>The range of RDQS: [0, 31]. | RW |

### 16.5.1.16 Toggle NAND DQS Hold Time for Data Input Finish (bank 1/2/3 TGHL)

|            |             |    |    |    |    |    |    |    |       |         |    |    |    |    |    |         |                   |    |    |    |    |         |   |   |   |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|-------|---------|----|----|----|----|----|---------|-------------------|----|----|----|----|---------|---|---|---|---|---|---|---|---|---|---|
|            | <b>TGHL</b> |    |    |    |    |    |    |    |       |         |    |    |    |    |    |         | <b>0x13410150</b> |    |    |    |    |         |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22      | 21 | 20 | 19 | 18 | 17 | 16      | 15                | 14 | 13 | 12 | 11 | 10      | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|            | Reserved    |    |    |    |    |    |    |    | CDQSD | CDQSH 3 |    |    |    |    |    | CDQSH 2 |                   |    |    |    |    | CDQSH 1 |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b> | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0     | 0       | 1  | 0  | 0  | 1  | 0  | 1       | 0                 | 0  | 1  | 0  | 0  | 1       | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

| Bits  | Name     | Description                                  | RW |
|-------|----------|----------------------------------------------|----|
| 31:25 | Reserved | Writing has no effect, read as zero.         | R  |
| 24    | CDQSD    | <b>DQS Hold Time for data input Finish</b>   | R  |
| 23:16 | CDQSH 3  | <b>DQS Hold Time for data input (bank3).</b> | RW |
| 15:8  | CDQSH 2  | <b>DQS Hold Time for data input (bank2).</b> | RW |
| 7:0   | CDQSH 1  | <b>DQS Hold Time for data input (bank1).</b> | RW |

### 16.5.1.17 Toggle NAND DQS Hold Time for Data Input Finish (bank 4/5/6 TGHH)

| TGHH |          |    |    |    |    |    |    |    |         |    |    |    |    |    |    | 0x13410154 |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
|------|----------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|------------|---------|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
| Bit  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|      | Reserved |    |    |    |    |    |    |    | CDQSH 6 |    |    |    |    |    |    |            | CDQSH 5 |    |    |    |    |    |   |   | CDQSH 4 |   |   |   |   |   |   |   |
| RST  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 1  | 0  | 0  | 1  | 0  | 1          | 0       | 0  | 1  | 0  | 0  | 1  | 0 | 1 | 0       | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

| Bits  | Name     | Description                                  | RW |
|-------|----------|----------------------------------------------|----|
| 31:24 | Reserved | Writing has no effect, read as zero.         | R  |
| 23:16 | CDQSH 6  | <b>DQS Hold Time for data input</b> (bank6). | RW |
| 15:8  | CDQSH 5  | <b>DQS Hold Time for data input</b> (bank5). | RW |
| 7:0   | CDQSH 4  | <b>DQS Hold Time for data input</b> (bank4). | RW |

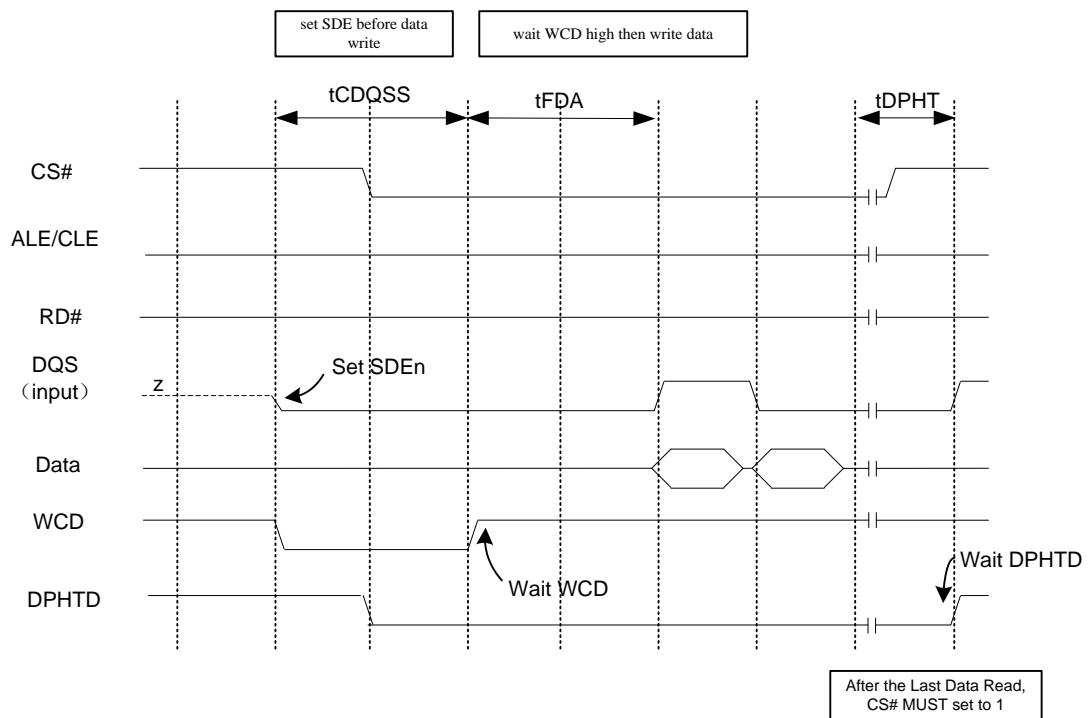


Figure 16-12 Basic Timing of Toggle NAND Write

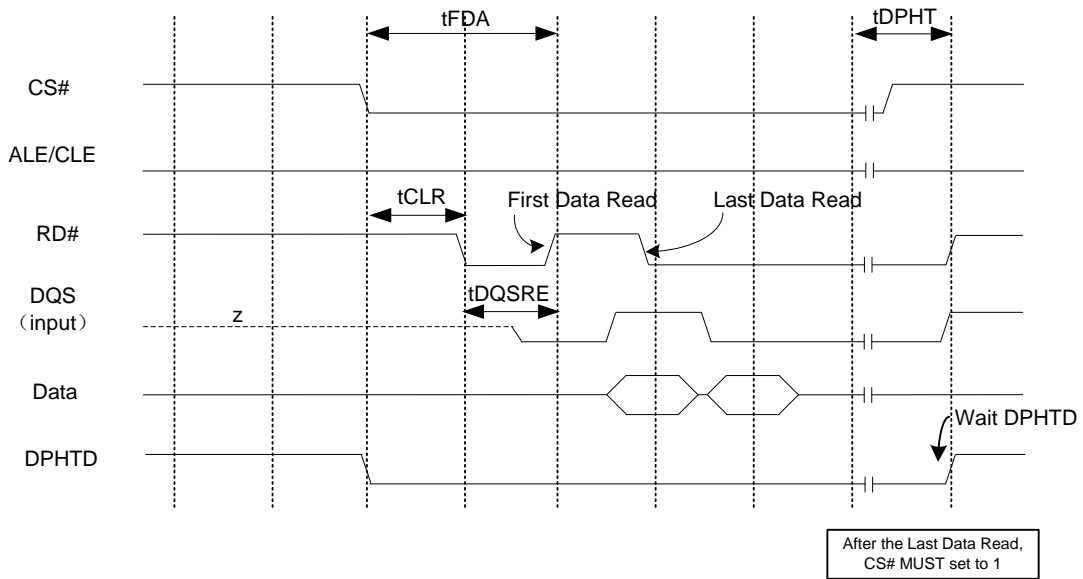


Figure 16-13 Basic Timing of Toggle NAND Read

NOTE.

- a)  $t_{FDA}$  MUST  $>$   $t_{CLR}$ ;
- b)  $t_{DQSRE}$  have nothing to do with  $t_{FDA}$ ;

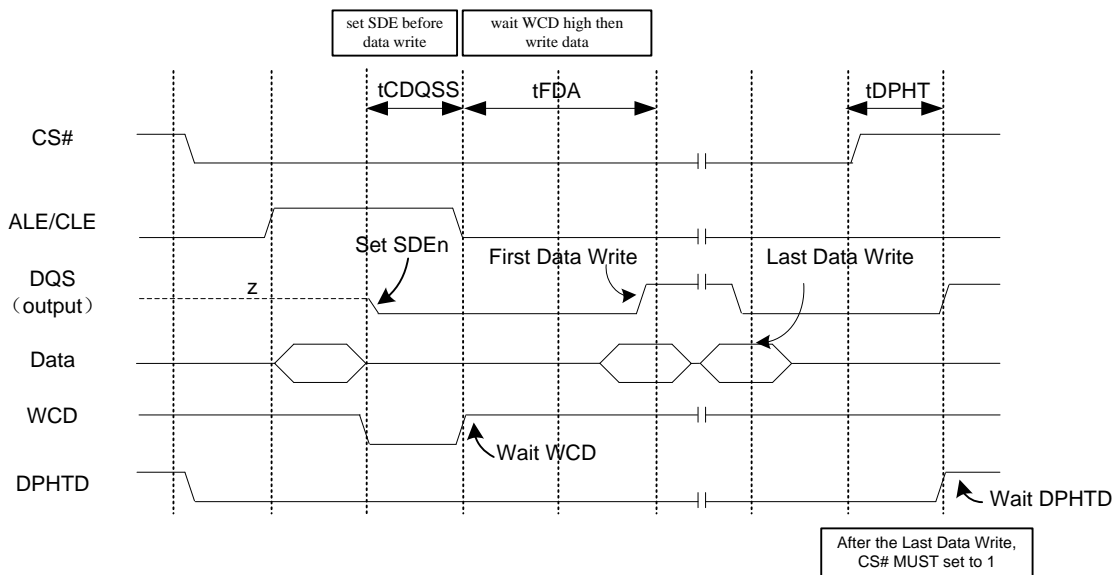


Figure 16-14 Basic Timing of Toggle NAND Page Write

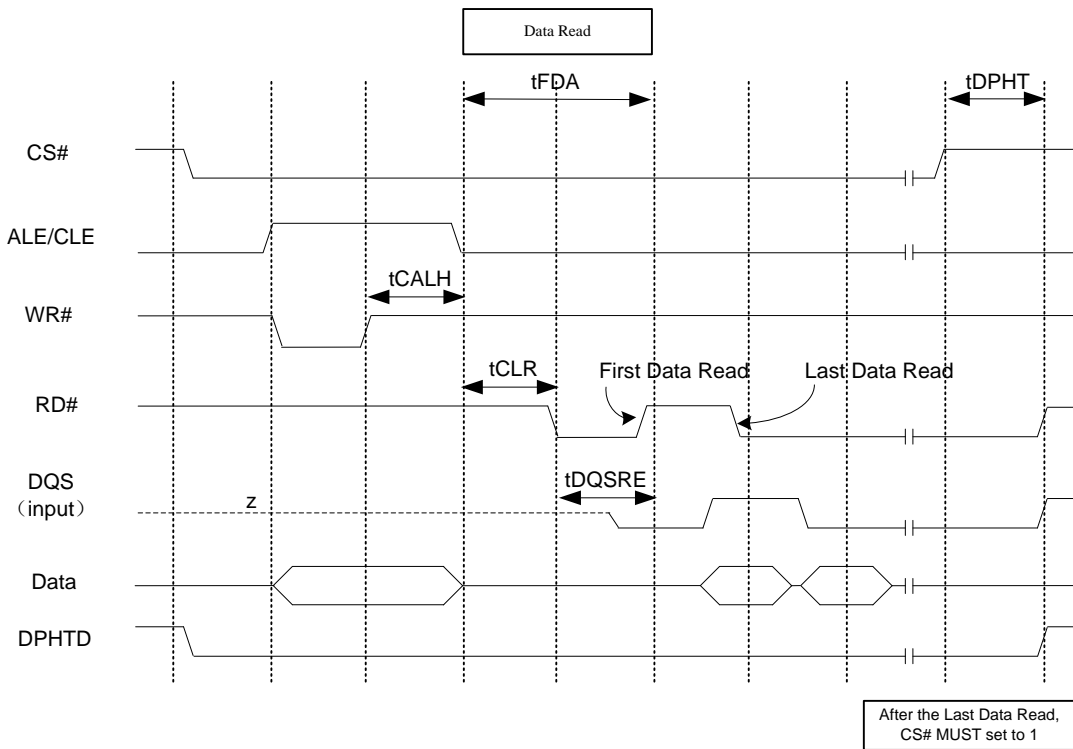


Figure 16-15 Basic Timing of Toggle NAND Page Read

### 16.5.2 NAND Flash Boot Loader

To support boot from NAND flash, 16KB on-chip Boot ROM is implemented. Following figure illustrates the structure of NAND Flash Boot Loader.

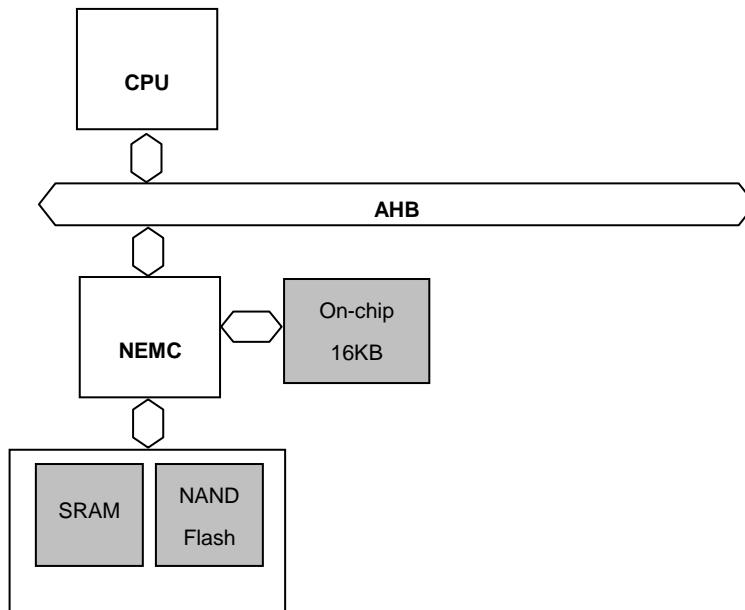


Figure 16-16 Structure of NAND Flash Boot Loader

When system is configured to boot from NAND flash, after reset, the program in Boot ROM is executed and the program will copy the first 8K bytes of NAND flash to internal memory for further initialization.

Generally, the boot code will copy more NAND flash content to DRAM. Then the main program will be executed on DRAM.

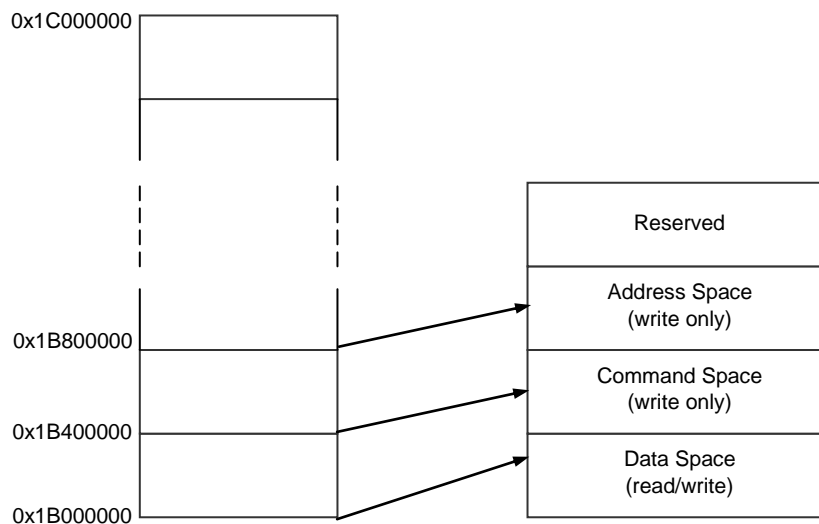
When system is configured to boot from NAND flash, software may know the nand flash page size through BOOT\_SEL[2:0] pin.

### 16.5.3 NAND Flash Operation

Set NFEn bit of NAND Flash Control/Status Register (NFCSR) will enable access to NAND flash. The partition of static bank n (n=1~6) is changed as following figure. Writes to any of address space will be translated to NAND flash address cycle. Writes to any of command space will be translated to NAND flash command cycle.

#### CAUTION:

1. Don't read to address and command space, and these two partitions should be uncacheable. Reads and writes to any of data space will be translated to NAND flash data read/write cycle. DMA access to data space is supported to increase the speed of data read/write. The DMA access cannot exceed the page boundary (512 bytes or 2K bytes) of NAND.
2. The least significant bit of the column address shall always be zero for a Toggle interface, i.e. an **EVEN** number of bytes is always transferred.



**Figure 16-17 Static Bank 1 Partition When NAND Flash is Used (an example)**

The timing of NAND flash access is configured by SMCRn and is same as normal static memory timing, except that CSn# is controlled by NFCE bit NFCSR. CSn# is always asserted when NFCE is 1. When NFCE is 0, CSn# is asserted as normal static memory access.

The control signals for direction connection of NAND flash are CSn#, FRE#, FWE#, FRB#(GPIO), A1 and A0. Following figure shows the connection between processor and NAND Flash.

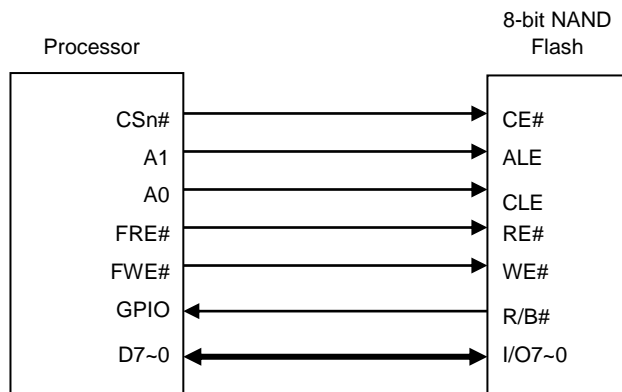


Figure 16-18 Example of 8-bit NAND Flash Connection

### 16.5.4 Example of Toggle NAND Flash Access

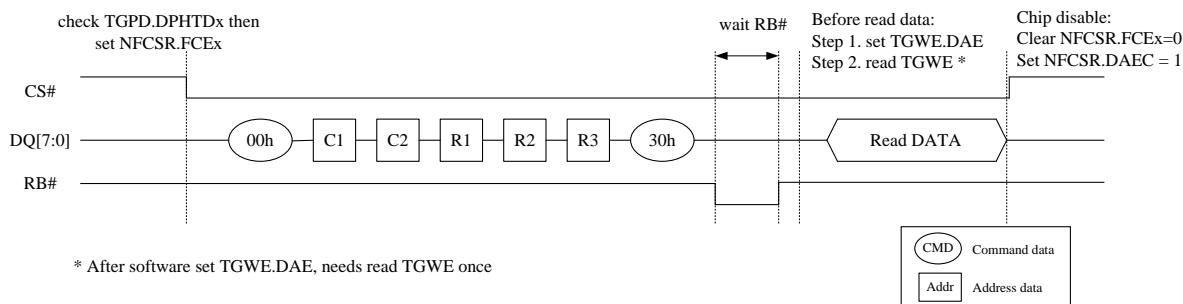


Figure 16-19 Toggle NAND Page Read Operation

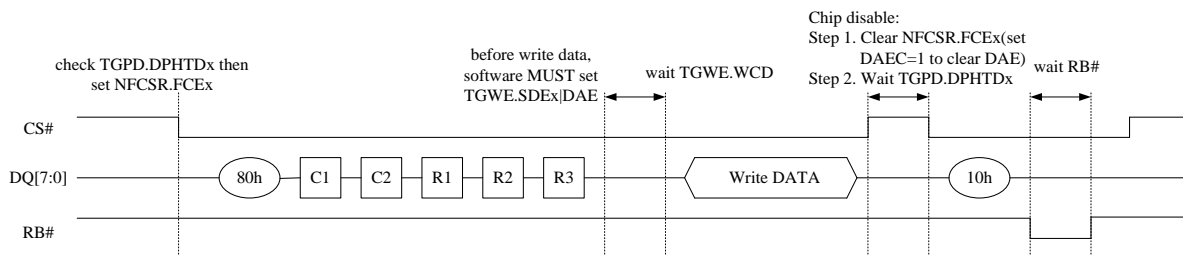


Figure 16-20 Toggle NAND Page Program Operation

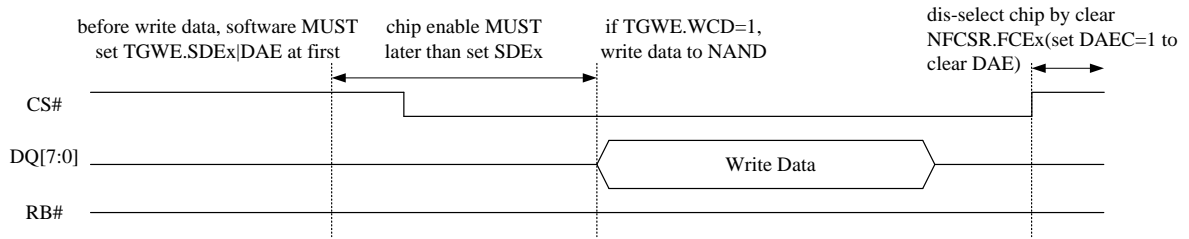


Figure 16-21 Toggle NAND Page Program Operation (Basic)

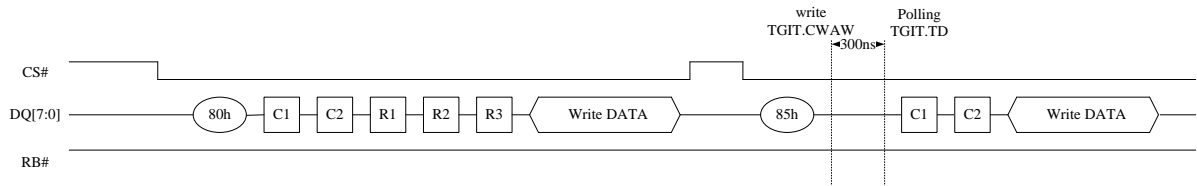


Figure 16-22 Program Operation with Random Data Input

# 17 BCH Controller

## 17.1 Overview

The BCH Controller implements data ECC encoding and decoding.

Table 17-1 BCH Registers

| Name    | Description                 | RW | Reset Value | Address    | Access Width |
|---------|-----------------------------|----|-------------|------------|--------------|
| BHCR    | BCH Control register        | RW | 0x00000000  | 0x134D0000 | 32           |
| BHCSR   | BCH Control Set register    | W  | Undefined   | 0x134D0004 | 32           |
| BHCCR   | BCH Control Clear register  | W  | Undefined   | 0x134D0008 | 32           |
| BHCNT   | BCH ENC/DEC Count register  | RW | 0x00000000  | 0x134D000C | 32           |
| BHDR    | BCH data register           | W  | Undefined   | 0x134D0010 | 32/16/8      |
| BHPAR0  | BCH Parity 0 register       | R  | 0x00000000  | 0x134D0014 | 32           |
| BHPAR1  | BCH Parity 1 register       | R  | 0x00000000  | 0x134D0018 | 32           |
| BHPAR2  | BCH Parity 2 register       | R  | 0x00000000  | 0x134D001C | 32           |
| BHPAR3  | BCH Parity 3 register       | R  | 0x00000000  | 0x134D0020 | 32           |
| BHPAR4  | BCH Parity 4 register       | R  | 0x00000000  | 0x134D0024 | 32           |
| BHPAR5  | BCH Parity 5 register       | R  | 0x00000000  | 0x134D0028 | 32           |
| BHPAR6  | BCH Parity 6 register       | R  | 0x00000000  | 0x134D002C | 32           |
| BHPAR7  | BCH Parity 7 register       | R  | 0x00000000  | 0x134D0030 | 32           |
| BHPAR8  | BCH Parity 8 register       | R  | 0x00000000  | 0x134D0034 | 32           |
| BHPAR9  | BCH Parity 9 register       | R  | 0x00000000  | 0x134D0038 | 32           |
| BHPAR10 | BCH Parity 10 register      | R  | 0x00000000  | 0x134D003C | 32           |
| BHPAR11 | BCH Parity 11 register      | R  | 0x00000000  | 0x134D0040 | 32           |
| BHPAR12 | BCH Parity 12 register      | R  | 0x00000000  | 0x134D0044 | 32           |
| BHPAR13 | BCH Parity 13 register      | R  | 0x00000000  | 0x134D0048 | 32           |
| BHPAR14 | BCH Parity 14 register      | R  | 0x00000000  | 0x134D004C | 32           |
| BHPAR15 | BCH Parity 15 register      | R  | 0x00000000  | 0x134D0050 | 32           |
| BHPAR16 | BCH Parity 16 register      | R  | 0x00000000  | 0x134D0054 | 32           |
| BHPAR17 | BCH Parity 17 register      | R  | 0x00000000  | 0x134D0058 | 32           |
| BHPAR18 | BCH Parity 18 register      | R  | 0x00000000  | 0x134D005C | 32           |
| BHPAR19 | BCH Parity 19 register      | R  | 0x00000000  | 0x134D0060 | 32           |
| BHPAR20 | BCH Parity 20 register      | R  | 0x00000000  | 0x134D0064 | 32           |
| BHPAR21 | BCH Parity 21 register      | R  | 0x00000000  | 0x134D0068 | 32           |
| BHPAR22 | BCH Parity 22 register      | R  | 0x00000000  | 0x134D006C | 32           |
| BHPAR23 | BCH Parity 23 register      | R  | 0x00000000  | 0x134D0070 | 32           |
| BHPAR24 | BCH Parity 24 register      | R  | 0x00000000  | 0x134D0074 | 32           |
| BHPAR25 | BCH Parity 25 register      | R  | 0x00000000  | 0x134D0078 | 32           |
| BHPAR26 | BCH Parity 26 register      | R  | 0x00000000  | 0x134D007C | 32           |
| BHPAR27 | BCH Parity 27 register      | R  | 0x00000000  | 0x134D0080 | 32           |
| BHERR0  | BCH Error Report 0 register | R  | 0x00000000  | 0x134D0084 | 32           |
| BHERR1  | BCH Error Report 1 register | R  | 0x00000000  | 0x134D0088 | 32           |
| BHERR2  | BCH Error Report 2 register | R  | 0x00000000  | 0x134D008C | 32           |
| BHERR3  | BCH Error Report 3 register | R  | 0x00000000  | 0x134D0090 | 32           |
| BHERR4  | BCH Error Report 4 register | R  | 0x00000000  | 0x134D0094 | 32           |
| BHERR5  | BCH Error Report 5 register | R  | 0x00000000  | 0x134D0098 | 32           |
| BHERR6  | BCH Error Report 6 register | R  | 0x00000000  | 0x134D009C | 32           |
| BHERR7  | BCH Error Report 7 register | R  | 0x00000000  | 0x134D00A0 | 32           |



|         |                              |   |            |            |    |
|---------|------------------------------|---|------------|------------|----|
| BHERR8  | BCH Error Report 8 register  | R | 0x00000000 | 0x134D00A4 | 32 |
| BHERR9  | BCH Error Report 9 register  | R | 0x00000000 | 0x134D00A8 | 32 |
| BHERR10 | BCH Error Report 10 register | R | 0x00000000 | 0x134D00AC | 32 |
| BHERR11 | BCH Error Report 11 register | R | 0x00000000 | 0x134D00B0 | 32 |
| BHERR12 | BCH Error Report 12 register | R | 0x00000000 | 0x134D00B4 | 32 |
| BHERR13 | BCH Error Report 13 register | R | 0x00000000 | 0x134D00B8 | 32 |
| BHERR14 | BCH Error Report 14 register | R | 0x00000000 | 0x134D00BC | 32 |
| BHERR15 | BCH Error Report 15 register | R | 0x00000000 | 0x134D00C0 | 32 |
| BHERR16 | BCH Error Report 16 register | R | 0x00000000 | 0x134D00C4 | 32 |
| BHERR17 | BCH Error Report 17 register | R | 0x00000000 | 0x134D00C8 | 32 |
| BHERR18 | BCH Error Report 18 register | R | 0x00000000 | 0x134D00CC | 32 |
| BHERR19 | BCH Error Report 19 register | R | 0x00000000 | 0x134D00D0 | 32 |
| BHERR20 | BCH Error Report 20 register | R | 0x00000000 | 0x134D00D4 | 32 |
| BHERR21 | BCH Error Report 21 register | R | 0x00000000 | 0x134D00D8 | 32 |
| BHERR22 | BCH Error Report 22 register | R | 0x00000000 | 0x134D00DC | 32 |
| BHERR23 | BCH Error Report 23 register | R | 0x00000000 | 0x134D00E0 | 32 |
| BHERR24 | BCH Error Report 24 register | R | 0x00000000 | 0x134D00E4 | 32 |
| BHERR25 | BCH Error Report 25 register | R | 0x00000000 | 0x134D00E8 | 32 |
| BHERR26 | BCH Error Report 26 register | R | 0x00000000 | 0x134D00EC | 32 |
| BHERR27 | BCH Error Report 27 register | R | 0x00000000 | 0x134D00F0 | 32 |
| BHERR28 | BCH Error Report 28 register | R | 0x00000000 | 0x134D00F4 | 32 |
| BHERR29 | BCH Error Report 29 register | R | 0x00000000 | 0x134D00F8 | 32 |
| BHERR30 | BCH Error Report 30 register | R | 0x00000000 | 0x134D00FC | 32 |
| BHERR31 | BCH Error Report 31 register | R | 0x00000000 | 0x134D0100 | 32 |
| BHERR32 | BCH Error Report 32 register | R | 0x00000000 | 0x134D0104 | 32 |
| BHERR33 | BCH Error Report 33 register | R | 0x00000000 | 0x134D0108 | 32 |
| BHERR34 | BCH Error Report 34 register | R | 0x00000000 | 0x134D010C | 32 |
| BHERR35 | BCH Error Report 35 register | R | 0x00000000 | 0x134D0110 | 32 |
| BHERR36 | BCH Error Report 36 register | R | 0x00000000 | 0x134D0114 | 32 |
| BHERR37 | BCH Error Report 37 register | R | 0x00000000 | 0x134D0118 | 32 |
| BHERR38 | BCH Error Report 38 register | R | 0x00000000 | 0x134D011C | 32 |
| BHERR39 | BCH Error Report 39 register | R | 0x00000000 | 0x134D0120 | 32 |
| BHERR40 | BCH Error Report 40 register | R | 0x00000000 | 0x134D0124 | 32 |
| BHERR41 | BCH Error Report 41 register | R | 0x00000000 | 0x134D0128 | 32 |
| BHERR42 | BCH Error Report 42 register | R | 0x00000000 | 0x134D012C | 32 |
| BHERR43 | BCH Error Report 43 register | R | 0x00000000 | 0x134D0130 | 32 |
| BHERR44 | BCH Error Report 44 register | R | 0x00000000 | 0x134D0134 | 32 |
| BHERR45 | BCH Error Report 45 register | R | 0x00000000 | 0x134D0138 | 32 |
| BHERR46 | BCH Error Report 46 register | R | 0x00000000 | 0x134D013C | 32 |
| BHERR47 | BCH Error Report 47 register | R | 0x00000000 | 0x134D0140 | 32 |
| BHERR48 | BCH Error Report 48 register | R | 0x00000000 | 0x134D0144 | 32 |
| BHERR49 | BCH Error Report 49 register | R | 0x00000000 | 0x134D0148 | 32 |
| BHERR50 | BCH Error Report 50 register | R | 0x00000000 | 0x134D014C | 32 |
| BHERR51 | BCH Error Report 51 register | R | 0x00000000 | 0x134D0150 | 32 |
| BHERR52 | BCH Error Report 52 register | R | 0x00000000 | 0x134D0154 | 32 |
| BHERR53 | BCH Error Report 53 register | R | 0x00000000 | 0x134D0158 | 32 |
| BHERR54 | BCH Error Report 54 register | R | 0x00000000 | 0x134D015C | 32 |
| BHERR55 | BCH Error Report 55 register | R | 0x00000000 | 0x134D0160 | 32 |
| BHERR56 | BCH Error Report 56 register | R | 0x00000000 | 0x134D0164 | 32 |
| BHERR57 | BCH Error Report 57 register | R | 0x00000000 | 0x134D0168 | 32 |
| BHERR58 | BCH Error Report 58 register | R | 0x00000000 | 0x134D016C | 32 |
| BHERR59 | BCH Error Report 59 register | R | 0x00000000 | 0x134D0170 | 32 |
| BHERR60 | BCH Error Report 60 register | R | 0x00000000 | 0x134D0174 | 32 |
| BHERR61 | BCH Error Report 61 register | R | 0x00000000 | 0x134D0178 | 32 |

|         |                               |    |            |            |    |
|---------|-------------------------------|----|------------|------------|----|
| BHERR62 | BCH Error Report 62 register  | R  | 0x00000000 | 0x134D017C | 32 |
| BHERR63 | BCH Error Report 63 register  | R  | 0x00000000 | 0x134D0180 | 32 |
| BHINT   | BCH Interrupt Status register | RW | 0x00000000 | 0x134D0184 | 32 |
| BHINTES | BCH Interrupt Set register    | W  | Undefined  | 0x134D0188 | 32 |
| BHINTEC | BCH Interrupt Clear register  | W  | Undefined  | 0x134D018C | 32 |
| BHINTE  | BCH Interrupt Enable register | RW | 0x00000000 | 0x134D0190 | 32 |
| BHTO    | BCH User Tag Output register  | R  | Undefined  | 0x134D0194 | 32 |

## 17.2 Register Description

### 17.2.1 BCH Control Register (BHCR)

BHCR is a 32-bit read/write register that is used to configure BCH controller. It is initialized by any reset.

|             |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |      |    |     |          |      |    |   |   |   |   |   |   |          |      |      |      |   |
|-------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|------|----|-----|----------|------|----|---|---|---|---|---|---|----------|------|------|------|---|
| <b>BHCR</b> |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x134D0000</b> |      |    |     |          |      |    |   |   |   |   |   |   |          |      |      |      |   |
| Bit         | 31  | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15   | 14 | 13  | 12       | 11   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3        | 2    | 1    | 0    |   |
|             | TAG |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   | MZEB |    | BPS | Reserved | BSEL |    |   |   |   |   |   |   | Reserved | ENCE | INIT | BCHE |   |
| RST         | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0    | 0  | 0   | 0        | 0    | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0        | 0    | 0    | 0    | 0 |

| Bits        | Name                              | Description                                                                                                                                                                                                                                                                                                                                                                    | RW          |                    |   |                                   |    |                     |    |
|-------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------------------|---|-----------------------------------|----|---------------------|----|
| 31:16       | TAG                               | <b>User-provided TAG:</b> Indicates a user-provided tag associated with the data block. The tag can be used to track data through the ECC module and the contents of the tag are presented on the BHTO.TAGO at the completion of the correction operation.                                                                                                                     | RW          |                    |   |                                   |    |                     |    |
| 15:13       | MZEB                              | <b>MAX ZERO BIT IN ERASED BLOCK:</b> Indicates that the maximum number of zero bits in an erased data block.<br>In decoding mode, if the number of zero bits less than MZEB, the BCH Controller will set BHINT.ALL_f to 1.                                                                                                                                                     | RW          |                    |   |                                   |    |                     |    |
| 12          | BPS                               | <b>DECODER BYPASS:</b> Indicates that no corrections were applied to the data block since the BPS was asserted at the beginning of the correction operation.<br><br><table border="0"> <tr> <td><b>BPS</b></td> <td><b>Description</b></td> </tr> <tr> <td>0</td> <td>Decoder is active (Initial value)</td> </tr> <tr> <td>1</td> <td>Decoder is bypassed</td> </tr> </table> | <b>BPS</b>  | <b>Description</b> | 0 | Decoder is active (Initial value) | 1  | Decoder is bypassed | RW |
| <b>BPS</b>  | <b>Description</b>                |                                                                                                                                                                                                                                                                                                                                                                                |             |                    |   |                                   |    |                     |    |
| 0           | Decoder is active (Initial value) |                                                                                                                                                                                                                                                                                                                                                                                |             |                    |   |                                   |    |                     |    |
| 1           | Decoder is bypassed               |                                                                                                                                                                                                                                                                                                                                                                                |             |                    |   |                                   |    |                     |    |
| 11          | Reserved                          | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                           | R           |                    |   |                                   |    |                     |    |
| 10:4        | BSEL                              | <b>BCH Encoding/Decoding Bit Select:</b> It is used to select the correction algorithm. Only multiples of 4 are supported (i.e. ECC 4, 8, 12 ... 56, 60, 64).<br><br><table border="0"> <tr> <td><b>BSEL</b></td> <td><b>Description</b></td> </tr> <tr> <td>4</td> <td>4-bit correction</td> </tr> </table>                                                                   | <b>BSEL</b> | <b>Description</b> | 4 | 4-bit correction                  | RW |                     |    |
| <b>BSEL</b> | <b>Description</b>                |                                                                                                                                                                                                                                                                                                                                                                                |             |                    |   |                                   |    |                     |    |
| 4           | 4-bit correction                  |                                                                                                                                                                                                                                                                                                                                                                                |             |                    |   |                                   |    |                     |    |

|   |          |                                                                                                                                                                                                                  |    |
|---|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|   |          | 8      8-bit correction<br>12     12-bit correction<br>...<br>60     60-bit correction<br>64     64-bit correction                                                                                               |    |
| 3 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                             | R  |
| 2 | ENCE     | <b>BCH Encoding/Decoding Select:</b> It is used to define whether in encoding or in decoding phase when BCH is used.<br><br><b>ENCE    Description</b><br>0        Decoding (Initial value)<br>1        Encoding | RW |
| 1 | INIT     | <b>BCH Controller Initial:</b> Initializes BCH Controller for a new operation. This bit is cleared automatically by hardware and always read as 0.                                                               | W  |
| 0 | BCHE     | <b>BCH Enable:</b> BCH correction is enable/disable.<br><br><b>BCHE    Description</b><br>0        BCH is disabled (initial value)<br>1        BCH is enabled                                                    | RW |

### 17.2.2 BCH Control Set Register (BHCSR)

BHCSR is a 32-bit write-only register that is used to set BCH controller to 1.

When write 1 to BHCSR, the corresponding bit in BHCR register is set to 1. Write 0 to BHCSR is ignored.

| BHCSR |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x134D0004 |       |      |          |       |    |    |   |   |   |   |   |   |   |   |   |          |       |       |       |   |
|-------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|-------|------|----------|-------|----|----|---|---|---|---|---|---|---|---|---|----------|-------|-------|-------|---|
| Bit   | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15    | 14   | 13       | 12    | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0        |       |       |       |   |
|       | TAGS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | MZEBS | BPSS | Reserved | BSELS |    |    |   |   |   |   |   |   |   |   |   | Reserved | ENCES | INITS | BCHES |   |
| RST   | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0     | 0    | 0        | 0     | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0        | 0     | 0     | 0     | 0 |

| Bits  | Name     | Description                                                                    | RW |
|-------|----------|--------------------------------------------------------------------------------|----|
| 31:16 | TAGS     | <b>User-provided TAG Set:</b> It is used to set BHCR.TAG to 1.                 | W  |
| 15:13 | MZEBS    | <b>MAX ZERO BIT IN ERASED BLOCK Set:</b> it is used to set BHCR.MZEB to 1.     | W  |
| 12    | BPSS     | <b>DECODER BYPASS Set:</b> it is used to set BHCR.BPS to 1                     | W  |
| 11    | Reserved | Writing has no effect, read as zero.                                           | R  |
| 10:4  | BSELS    | <b>BCH Encoding/Decoding Bit Select Set:</b> It is used to set BHCR.BSEL to 1. | W  |
| 3     | Reserved | Writing has no effect, read as zero.                                           | R  |
| 2     | ENCES    | <b>BCH Encoding/Decoding Select Set:</b> It is used to set BHCR.ENCE to 1.     | W  |

|   |       |                                                                      |   |
|---|-------|----------------------------------------------------------------------|---|
| 1 | INITS | <b>BCH Controller Initial Set:</b> It is used to set BHCR.INIT to 1. | W |
| 0 | BCHE  | <b>BCH Enable Set:</b> It is used to set BHCR.BCHE to 1.             | W |

### 17.2.3 BCH Control Clear Register (BHCCR)

BHCCR is a 32-bit write-only register that is used to clear BCH controller to 0.

When write 1 to BHCCR, the corresponding bit in BHCR register is cleared to 0. Write 0 to BHCCR is ignored.

| BHCCR |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x134D0008 |      |    |      |          |       |    |   |   |   |   |   |   |          |      |       |       |   |
|-------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|------|----|------|----------|-------|----|---|---|---|---|---|---|----------|------|-------|-------|---|
| Bit   | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15   | 14 | 13   | 12       | 11    | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3        | 2    | 1     | 0     |   |
|       | TAGC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | MZEB |    | BPCS | Reserved | BSELC |    |   |   |   |   |   |   | Reserved | ENEC | INITC | BCHEC |   |
| RST   | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0    | 0  | 0    | 0        | 0     | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0        | 0    | 0     | 0     | 0 |

| Bits  | Name     | Description                                                                        | RW |
|-------|----------|------------------------------------------------------------------------------------|----|
| 31:16 | TAGC     | <b>User-provided TAG Clear:</b> It is used to clear BHCR.TAG to 0.                 | W  |
| 15:13 | MZEB     | <b>MAX ZERO BIT IN ERASED BLOCK Clear:</b> It is used to clear BHCR.MZEB to 0.     | W  |
| 12    | BPCS     | <b>DECODER BYPASS Clear:</b> It is used to clear BHCR.BPS to 0                     | W  |
| 11    | Reserved | Writing has no effect, read as zero.                                               | R  |
| 10:4  | BSELC    | <b>BCH Encoding/Decoding Bit Select Clear:</b> It is used to clear BHCR.BSEL to 0. | W  |
| 3     | Reserved | Writing has no effect, read as zero.                                               | R  |
| 2     | ENEC     | <b>BCH Encoding/Decoding Select Clear:</b> It is used to clear BHCR.ENEC to 0.     | W  |
| 1     | INITC    | <b>BCH Controller Initial Clear:</b> It is used to clear BHCR.INIT to 0.           | W  |
| 0     | BCHEC    | <b>BCH Enable Clear:</b> It is used to clear BHCR.BCHE to 0.                       | W  |

### 17.2.4 BCH ENC/DEC Count Register (BHCNT)

BHCNT is a 32-bit read/write register that is used to indicate the total number of 8-bit data during encoding or decoding. It is initialized by any reset.

| BHCNT |          |    |    |    |    |    |    |    |            |    |    |    |    |    |    | 0x134D000C |          |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |   |
|-------|----------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|------------|----------|----|----|----|------------|----|---|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23         | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15       | 14 | 13 | 12 | 11         | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|       | Reserved |    |    |    |    |    |    |    | PARITYSIZE |    |    |    |    |    |    |            | Reserved |    |    |    | BLOCK SIZE |    |   |   |   |   |   |   |   |   |   |   |   |
| RST   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0        | 0  | 0  | 0  | 0          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name       | Description                                                                                                                                                                                                                           | RW |
|-------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:23 | Reserved   | Writing has no effect, read as zero.                                                                                                                                                                                                  | R  |
| 22:16 | PARITYSIZE | <b>Parity size:</b> The number of parity BYTE generated is $14 \cdot ecc\_level / 8$ .<br><b>Example:</b><br>If BHCR.BSEL = 64, then parity size = $14 \cdot BHCR.BSEL / 8 = 112\text{Byte}$                                          | RW |
| 15:11 | Reserved   | Writing has no effect, read as zero.                                                                                                                                                                                                  | R  |
| 10:0  | BLOCKSIZE  | <b>Block Size:</b> Indicates the number of bytes in the data payload for an operation. Supports a wide range of block size (2-1900 bytes), but the size must be a multiple of two bytes. The block size ONLY describes the data size. | RW |

### 17.2.5 BCH Data Register (BHDR)

BHDR is a write-only register that is used to transfer ECC data to BCH.

Data on BHDR is in little-endian format.

|      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| BHDR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x134D0010 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| Bit  | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
| BHDR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 17.2.6 BCH Parity Register (BHPARn, n=0, 1, 2, 3, 4, 5...26, 27)

BHPARn (n=0, 1, 2, 3, 4... 26, 27) are all 32-bit read/write register that contains the encoding parity data during BCH correction.

Data on BHPARn is in little-endian format with the least-significant-byte (LSB) on bits [15:0] and the most-significant-byte (MSB) on bits [31:16].

Example:

When 64-bit BCH is selected, BHPAR0~BHPAR27 consist of the 896 ( $14 \times 64$ ) bits of parity data and bit 0 of BHPAR0 is the 896th bit of parity data and bit 0 of BHPAR27 is the 1st bit of parity data.

When 60-bit BCH is selected, BHPAR0~BHPAR26 consist of the 840 bits of parity data and bit 0 of BHPAR0 is the 840th bit of parity data and bit 7 of BHPAR26 is the 1st bit of parity data.

When 8-bit BCH is selected, BHPAR0~BHPAR3 consist of the 112 bits of parity data and bit 0 of BHPAR0 is the 112th bit of parity data and bit 15 of BHPAR3 is the 1st bit of parity data.

|     |                |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|-----|----------------|----|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|
|     | BHPAR0         |    | 0x134D0014 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | BHPAR1         |    | 0x134D0018 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | BHPAR2         |    | 0x134D001C |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | BHPAR3         |    | 0x134D0020 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | BHPAR4         |    | 0x134D0024 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | BHPAR5         |    | 0x134D0028 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | BHPAR6         |    | 0x134D002C |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | ...            |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | BHPAR25        |    | 0x134D0078 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | BHPAR26        |    | 0x134D007C |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|     | BHPAR27        |    | 0x134D0080 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| Bit | 31             | 30 | 29         | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |
|     | BHPAR0~BHPAR27 |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| RST | 0              | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 17.2.7 BCH Error Report Register (BHERRn, n=0,1,2,3,4,5,6,7,...,61,62,63)

BHERRn is 32-bit read/write register that contains the index for each error after BCH decoding. The correction index indicates the location of the correction with in the block. The index is a halfword index since the associated correction mask contains a 16-bits mask. The correction mask provides a 16bit mask that should be applied to the data at the associated index to correct the data. Any bit errors found in the parity region will NOT be reported as a correction (INDEX), but these bits will be reported in the number of errors status field (BHINT.TERRC).

|     |                         |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|-----|-------------------------|----|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|--------------------------|----|---|---|---|---|---|---|---|---|---|---|---|
|     | BHERR0                  |    | 0x134D0084 |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR1                  |    | 0x134D0088 |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR2                  |    | 0x134D008C |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR3                  |    | 0x134D0090 |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR4                  |    | 0x134D0094 |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR5                  |    | 0x134D0098 |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR6                  |    | 0x134D009C |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR7                  |    | 0x134D00A0 |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | ...                     |    | ...        |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR61                 |    | 0x134D0178 |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR62                 |    | 0x134D017C |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
|     | BHERR63                 |    | 0x134D0180 |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |                          |    |   |   |   |   |   |   |   |   |   |   |   |
| Bit | 31                      | 30 | 29         | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11                       | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|     | MASKn(n=0,1,2,...62,63) |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    | INDEXn(n=0,1,2,...62,63) |    |   |   |   |   |   |   |   |   |   |   |   |
| RST | 0                       | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0                        | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name  | Description                                                                                                                                                                                                                        | RW |
|-------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:16 | MASKn | <b>Error Byte MASK:</b> The correction mask provides a bitmask that should be applied to the data at the associated index to correct the data.<br>Example,<br>if n = 1, INDEX1 = 2, MASK1=0x5080, it means the second half word is | R  |

|       |          |                                                                                                                                                                                                                                                                                    |   |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|       |          | an error-halfword (three-error-bits, because 0x5080 have three bits 1 in binary) and needs to be XOR MASK1 (0x5080).                                                                                                                                                               |   |
| 15:11 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                               | R |
| 10:0  | INDEXn   | <b>Error Byte Index:</b> It is used to indicate the location of the error Byte.<br>Example,<br>INDEXn=0, it means the 1st halfword is an error halfword.<br>INDEXn=1, it means the 2nd halfword is an error halfword.<br>INDEXn=2, it means the 3rd halfword is an error halfword. | R |

### 17.2.8 BCH Interrupt Status Register (BHINT)

BHINT is a 32-bit read-only register that contains the interrupt flag and error count information during BCH correction. It is initialized by any reset. Software writes **1** to clear the corresponding bit except ERRC\BPSO\TERRC.

NOTE.

{DECF,UNCOR,ERR}=3'b100, means no error need be correct;

{DECF,UNCOR,ERR}=3'b11x, means uncorrectable error;

{DECF,UNCOR,ERR}=3'b101, means error can be correct;

| BHINT |          |      |    |    |    |    |    |    |    |      |       |    |    |    |    | 0x134D0184 |    |    |          |    |    |    |   |   |   |   |      |       |      |      |       |     |   |   |   |   |   |   |   |
|-------|----------|------|----|----|----|----|----|----|----|------|-------|----|----|----|----|------------|----|----|----------|----|----|----|---|---|---|---|------|-------|------|------|-------|-----|---|---|---|---|---|---|---|
| Bit   | 31       | 30   | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22   | 21    | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13       | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5    | 4     | 3    | 2    | 1     | 0   |   |   |   |   |   |   |   |
|       | Reserved | ERRC |    |    |    |    |    |    |    | BPSO | TERRC |    |    |    |    |            |    |    | Reserved |    |    |    |   |   |   |   | SDMF | ALL_f | DECF | ENCF | UNCOR | ERR |   |   |   |   |   |   |   |
| RST   | 0        | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0     | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0        | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0    | 0     | 0    | 0    | 0     | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW |
|-------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                          | R  |
| 30:24 | ERRC     | <b>Error Count:</b> It indicates the number of error <b>halfword</b> (16bits) in the data block. Any error found in the parity region will NOT be reported in this counter.<br>NOTE: ERRC is only valid when UNCOR = 0 and ERR = 1.<br><b>ERRC Description</b><br>0 No errors (Initial value)<br>1 One error-half-word in the data block<br>2 Two error-half-words in the data block<br>3 Three error-half-words in the data block<br>...<br>64 Sixty-four error-half-words in the data block | RW |
| 23    | BPSO     | <b>Bypass output:</b> Indicates that no corrections were applied to the data block since the BPSO was asserted at the beginning of the correction operation.                                                                                                                                                                                                                                                                                                                                  | RW |

|       |          |                                                                                                                                                                                                                                                                                                                                                                                                                       |    |
|-------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 22:16 | TERRC    | <p><b>Total Error Count:</b> It indicates the total number of error <b>bits</b> in the data block and parity region.</p> <p>NOTE: TERRC is only valid when UNCOR = 0 and ERR = 1.</p> <p><b>TERRC Description</b></p> <p>0 No errors (Initial value)</p> <p>1 One error in the data block</p> <p>2 Two errors in the data block</p> <p>3 Three errors</p> <p>4 Four errors</p> <p>...</p> <p>64 Sixty-four errors</p> | RW |
| 15:6  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                  | R  |
| 5     | SDMF     | <p><b>Decoding SYNDROME Finish:</b> It indicates that hardware finish Syndrome generating and only in decoding mode this bit will be set.</p> <p><b>SDMF Description</b></p> <p>0 Syndrome not Finish (Initial value)</p> <p>1 Syndrome Finish</p>                                                                                                                                                                    | RW |
| 4     | ALL_f    | <p><b>ALL_f:</b> It indicates that all data received during decoding are 0xff. When receiving all 0xff data, BCH doesn't correct the data and no error occurs.</p> <p><b>ALL_f Description</b></p> <p>0 Not all data (data + parity bytes) are 0xff (Initial value)</p> <p>1 All data (data + parity bytes) are 0xff</p>                                                                                              | RW |
| 3     | DECF     | <p><b>Decoding Finish:</b> It indicates that hardware finish BCH decoding.</p> <p><b>DECF Description</b></p> <p>0 Decoding not Finish (Initial value)</p> <p>1 Decoding Finish</p>                                                                                                                                                                                                                                   | RW |
| 2     | ENCF     | <p><b>Encoding Finish:</b> It indicates that hardware finish BCH encoding.</p> <p><b>ENCF Description</b></p> <p>0 Encoding not Finish (Initial value)</p> <p>1 Encoding Finish</p>                                                                                                                                                                                                                                   | RW |
| 1     | UNCOR    | <p><b>Un-correction Error:</b> Indicates that the block could not be corrected because the number of errors present is greater than the ECC capability.</p> <p><b>UNCOR Description</b></p> <p>0 No uncorrectable error (Initial value)</p> <p>1 Uncorrectable error occur</p>                                                                                                                                        | RW |
| 0     | ERR      | <p><b>Error:</b> It indicates that hardware detects error bits in data in the data block and parity block during BCH decoding.</p> <p><b>ERR Description</b></p> <p>0 No error (Initial value)</p> <p>1 Error occur</p>                                                                                                                                                                                               | RW |



### 17.2.9 BCH Interrupt Enable Set Register (BHINTES)

BHINTES is a 32-bit write-only register that is used to set BHINTE register. Writing 1 to BHINTES will set the corresponding bit in BHINTE to 1. Writing 0 to BHINTES is ignored.

| BHINTES |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x134D0188 |        |         |         |        |         |       |   |   |   |   |   |   |   |   |   |   |
|---------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|--------|---------|---------|--------|---------|-------|---|---|---|---|---|---|---|---|---|---|
| Bit     | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15     | 14      | 13      | 12     | 11      | 10    | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|         | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | SDMFES | ALL_FES | DECFCES | ENCFES | UNCORES | ERRES |   |   |   |   |   |   |   |   |   |   |
| RST     | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0      | 0       | 0       | 0      | 0       | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                           | RW |
|------|----------|---------------------------------------------------------------------------------------|----|
| 31:6 | Reserved | Writing has no effect, read as zero.                                                  | R  |
| 5    | SDMFES   | <b>SYNDROME Finish Interrupt Enable Set:</b> It is used to set BHINTE.SDMFE to 1.     | W  |
| 4    | ALL_FES  | <b>ALL_F Interrupt Enable Set:</b> It is used to set BHINTE.ALL_FE to 1.              | W  |
| 3    | DECFCES  | <b>Decoding Finish Interrupt Enable Set:</b> It is used to set BHINTE.DECFE to 1.     | W  |
| 2    | ENCFES   | <b>Encoding Finish Interrupt Enable Set:</b> It is used to set BHINTE.ENCFE to 1.     | W  |
| 1    | UNCORES  | <b>Un-correction Error Interrupt Enable Set:</b> It is used to set BHINTE.ENCFE to 1. | W  |
| 0    | ERRES    | <b>Error Interrupt Enable Set:</b> It is used to set BHINTE.ERRE to 1.                | W  |

### 17.2.10 BCH Interrupt Enable Clear Register (BHINTEC)

BHINTEC is a 32-bit write-only register that is used to clear BHINTE register. Writing 1 to BHINTEC will clear the corresponding bit in BHINTE to 0. Writing 0 to BHINTEC is ignored.

| BHINTEC |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x134D018C |        |         |         |        |        |       |   |   |   |   |   |   |   |   |   |   |
|---------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|--------|---------|---------|--------|--------|-------|---|---|---|---|---|---|---|---|---|---|
| Bit     | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15     | 14      | 13      | 12     | 11     | 10    | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|         | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | SDMFEC | ALL_FEC | DECFCFC | ENCFEC | UNCORC | ERREC |   |   |   |   |   |   |   |   |   |   |
| RST     | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0      | 0       | 0       | 0      | 0      | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                           | RW |
|------|----------|---------------------------------------------------------------------------------------|----|
| 31:5 | Reserved | Writing has no effect, read as zero.                                                  | R  |
| 5    | SDMFEC   | <b>SYNDROME Finish interrupt Enable Clear:</b> It is used to clear BHINTE.SDMFE to 0. | W  |
| 4    | ALL_FEC  | <b>ALL_F Interrupt Enable Clear:</b> It is used to clear BHINTE.ALL_FE to 0.          | W  |

|   |         |                                                                                            |   |
|---|---------|--------------------------------------------------------------------------------------------|---|
| 3 | DEC FEC | <b>Decoding Finish Interrupt Enable Clear:</b> It is used to clear BHINTE.DEC FE to 0.     | W |
| 2 | ENC FEC | <b>Encoding Finish Interrupt Enable Clear:</b> It is used to clear BHINTE.ENC FE to 0.     | W |
| 1 | UNCOREC | <b>Un-correction Error Interrupt Enable Clear:</b> It is used to clear BHINTE.ENC FE to 0. | W |
| 0 | ERREC   | <b>Error Interrupt Enable Clear:</b> It is used to set BHINTE.ERRE to 0.                   | W |

### 17.2.11 BCH Interrupt Enable Register (BHINTE)

BHINTE is a 32-bit read/write register that is used to enable/disable interrupts during BCH correction. It is initialized by any reset.

|               |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |       |        |        |        |        |      |   |   |   |   |   |   |   |   |   |   |
|---------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|-------|--------|--------|--------|--------|------|---|---|---|---|---|---|---|---|---|---|
| <b>BHINTE</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x134D0190</b> |       |        |        |        |        |      |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15    | 14     | 13     | 12     | 11     | 10   | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|               | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   | SDMFE | ALL_FE | DEC FE | ENC FE | UNCORE | ERRE |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0     | 0      | 0      | 0      | 0      | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |

| Bits | Name     | Description                                                                                                                                                                                                                   | RW |
|------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:6 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                          | R  |
| 5    | SDMFE    | <b>SYNDROME Finish Interrupt Enable:</b> It is used to enable or disable syndrome finish interrupt.<br><b>SDMFE Description</b><br>0 Disable Syndrome Finish interrupt(Initial value)<br>1 Enable Syndrome Finish interrupt   | RW |
| 4    | ALL_FE   | <b>ALL_F Interrupt Enable:</b> It is used to enable or disable all_f data interrupt.<br><b>ALL_FE Description</b><br>0 Disable ALL_F data interrupt (Initial value)<br>1 Enable ALL_F data interrupt                          | RW |
| 3    | DEC FE   | <b>Decoding Finish Interrupt Enable:</b> It is used to enable or disable decoding finish interrupt.<br><b>DEC FE Description</b><br>0 Disable Decoding Finish Interrupt (Initial value)<br>1 Enable Decoding Finish Interrupt | RW |
| 2    | ENC FE   | <b>Encoding Finish Interrupt Enable:</b> It is used to enable or disable encoding finish interrupt.<br><b>ENC FE Description</b><br>0 Disable Encoding Finish Interrupt (Initial value)<br>1 Enable Encoding Finish Interrupt | RW |

| 1      | UNCORE                                                | <b>Un-correction Error Interrupt Enable:</b> It is used to enable or disable un-correction error interrupt.<br><table border="1"> <thead> <tr> <th>UNCORE</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable Uncorrectable Error interrupt (Initial value)</td> </tr> <tr> <td>1</td> <td>Enable Uncorrectable Error Interrupt</td> </tr> </tbody> </table> | UNCORE | Description | 0 | Disable Uncorrectable Error interrupt (Initial value) | 1 | Enable Uncorrectable Error Interrupt | RW |
|--------|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------------|---|-------------------------------------------------------|---|--------------------------------------|----|
| UNCORE | Description                                           |                                                                                                                                                                                                                                                                                                                                                                                       |        |             |   |                                                       |   |                                      |    |
| 0      | Disable Uncorrectable Error interrupt (Initial value) |                                                                                                                                                                                                                                                                                                                                                                                       |        |             |   |                                                       |   |                                      |    |
| 1      | Enable Uncorrectable Error Interrupt                  |                                                                                                                                                                                                                                                                                                                                                                                       |        |             |   |                                                       |   |                                      |    |
| 0      | ERRE                                                  | <b>Error Interrupt Enable:</b> It is used to enable or disable error interrupt.<br><table border="1"> <thead> <tr> <th>ERRE</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Disable Error interrupt (Initial value)</td> </tr> <tr> <td>1</td> <td>Enable Error interrupt</td> </tr> </tbody> </table>                                                           | ERRE   | Description | 0 | Disable Error interrupt (Initial value)               | 1 | Enable Error interrupt               | RW |
| ERRE   | Description                                           |                                                                                                                                                                                                                                                                                                                                                                                       |        |             |   |                                                       |   |                                      |    |
| 0      | Disable Error interrupt (Initial value)               |                                                                                                                                                                                                                                                                                                                                                                                       |        |             |   |                                                       |   |                                      |    |
| 1      | Enable Error interrupt                                |                                                                                                                                                                                                                                                                                                                                                                                       |        |             |   |                                                       |   |                                      |    |

### 17.2.12 BCH User TAG OUTPUT Register (BHTO)

BHTO is a 32-bit read only register that is used to indicate the tag for the current corrected data block.

|             |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>BHTO</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x134D0194</b> |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   | TAGO |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | x    | x  | x  | x  | x  | x  | x | x | x | x | x | x | x | x | x | x |

| Bits  | Name     | Description                                                                                   | RW |
|-------|----------|-----------------------------------------------------------------------------------------------|----|
| 31:16 | Reserved | Writing has no effect, read as zero.<br>NOTE. Software must <b>NOT</b> write 1 to this field. | R  |
| 15:0  | TAGO     | <b>User-provided Tag OUTPUT:</b> Indicates the tag for the current corrected data block.      | R  |

## 17.3 BCH Operation

BCH controller supports a wide range of data payloads (2-1900 bytes), but the payload must be a multiple of two bytes. Only even ECC multiples are supported (i.e. ECC 2, 4, 6, 8 ... 58, 60, 62, 64). During encoding, hardware will generate 896-bit (14x64) parity data in 64-bit correction, 840-bit parity data in 60-bit correction, 560-bit parity data in 40-bit correction, 336-bit parity data in 24-bit correction, 224-bit parity data in 16-bit correction or 112-bit parity data in 8-bit correction, etc.. Parity data can be read out by CPU or DMA.

During decoding, if there are error bits in data block, after decoding BHERRn registers will hold the error location that can be read by CPU or DMA.

### 17.3.1 Encoding Sequence

BCH encoding can be operated by CPU or DMA.

### 17.3.1.1 CPU

- 1 Set BHCR.BCHE to 1 to enable BCH controller.
- 2 Select correction mode by setting BHCR.BSEL (64-bit, 60-bit... 8-bit, 4-bit or 2-bit).
- 3 Set BHCR.ENCE to 1 to enable encoding.
- 4 Set BHCNT to data block size and parity size in bytes.
- 5 Set BHCR.INIT to 1 to start a new operation.
- 6 Writes all data blocks to BHDR.
- 7 Check BHINTS.ENCF bit or by enabling encoding finish interrupt.
- 8 When encoding finishes, read out the parity data in BHPAR n.
- 9 After parity data is read out, clear BHINT.ENCF.

### 17.3.1.2 DMA

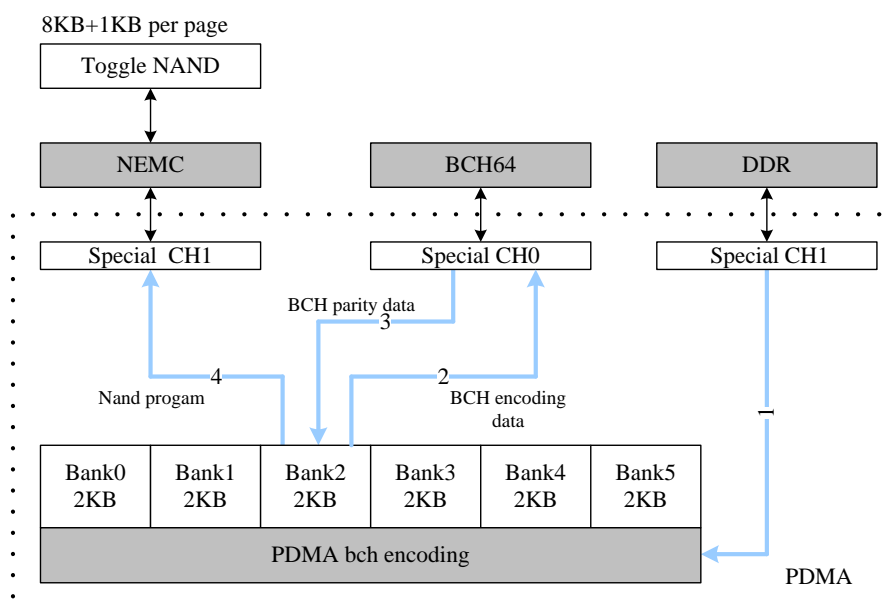


Figure 17-1 Block Diagram for BCH encoding

- 1 Set BHCR.BCHE to 1 to enable BCH controller.
- 2 Select correction mode by setting BHCR.BSEL (64-bit, 60-bit... 8-bit, 4-bit or 2-bit).
- 3 Set BHCR.ENCE to 1 to enable encoding.
- 4 Set BHCNT to data block size and parity size in bytes.
- 5 Set BHCR.INIT to 1 to start a new operation.
- 6 Start DMA transfer after configuring DMA channel.
- 7 DMA read data block from system memory and write to BCH controller automatically.
- 8 DMA will wait BCH *encoding* request when finishes writing data block.
- 9 BCH controller will issue encoding request to DMA when encoding ends.
- 10 DMA start to read out parity data.
- 11 After parity data is read out, DMA automatically clear BHINT.ENCF.

**NOTE:** When DMA is enabled, software should guarantee not to enable encoding finish interrupt.

### 17.3.2 Decoding Sequence

BCH decoding can be operated by CPU or DMA.

#### 17.3.2.1 CPU

- 1 Set BHCR.BCHE to 1 to enable BCH controller.
- 2 Select correction mode by setting BHCR.BSEL (64-bit, 60-bit... 8-bit, 4-bit or 2-bit).
- 3 Clear BHCR.ENCE to 0 to enable decoding.
- 4 Set BHCNT to data block size and parity size in bytes.
- 5 Set BHCR.INIT to 1 to start a new operation.
- 6 Write all data block to BHDR.
- 7 If another block needs to be decoded, please check BHINT.SDMF or by enabling syndrome finish interrupt. If BHINT.SDMF is asserted and another block needs to be decoded, set this bit to 1 to clear interrupt and go to step 5 to start a new decoding operation.
- 8 Check BHINTS.DECF bit or by enabling decoding finish interrupt.
- 9 When decoding finishes, read out the status in BHINT and error report in BHERRn.
- 10 After status and error report data is read out, set BHINT.DECF to 1.

#### 17.3.2.2 DMA

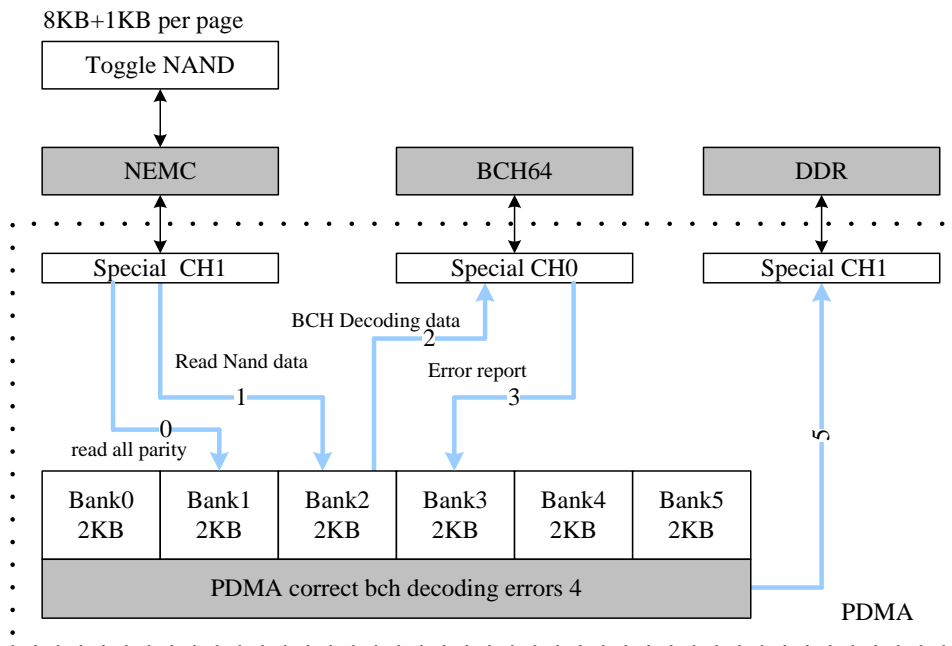


Figure 17-2 Block Diagram for BCH decoding

- 1 Set BHCR.BCHE to 1 to enable BCH controller.
- 2 Select correction mode by setting BHCR.BSEL (64-bit, 60-bit... 8-bit, 4-bit or 2-bit).
- 3 Clear BHCR.ENCE to 0 to enable decoding.
- 4 Set BHCNT to data block and parity size in bytes.
- 5 Set BHCR.INIT to 1 to start a new operation.
- 6 Start DMA transfer after configuring DMA channel.
- 7 DMA read data block from system memory and write to BCH controller automatically.
- 8 DMA will wait BCH *syndrome* or *decoding* request when finishes writing data block. (If *syndrome* request is asserted and another block needs to be decoded, set BHINT. SDMF to 1 to clear request, then go to step 6 to start a new decoding operation.)
- 9 BCH controller will issue decoding request to DMA when decoding ends.
- 10 DMA start to read out bch interrupt status (BHINT) and error report data (BHERRn) and write to memory.
- 11 After status and error report data is read out, set BHINT.DECF to 1 to clear *decoding* request.

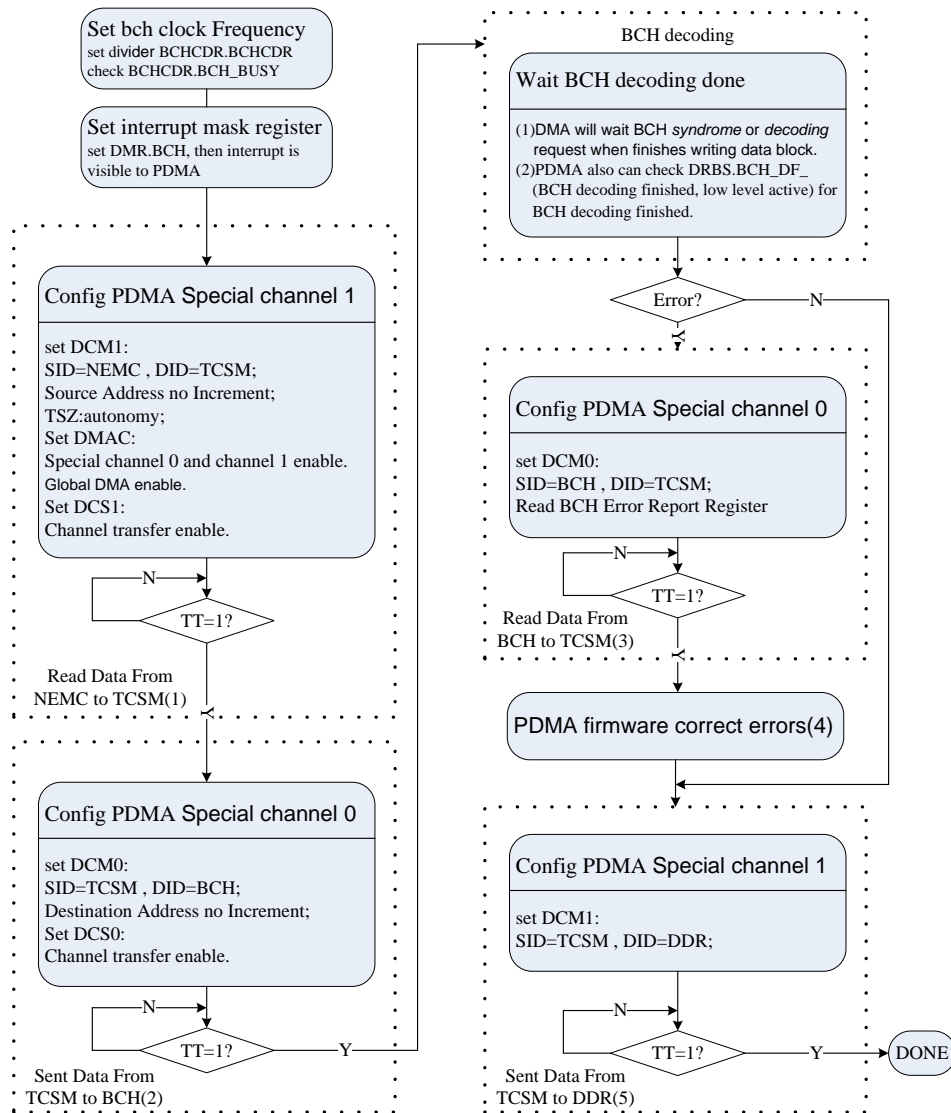


Figure 17-3 BCH decoding data flow

## Section 5

# SYSTEM FUNCTIONS

# 18 Clock Reset and Power Controller

## 18.1 Overview

The Clock & Power management block consists of three parts: Clock control, PLL control, and Power control, Reset control.

The Clock control logic can generate the required clock signals including CCLK for CPU, HHCLK for L2CACHE, H0CLK for the AHB0, H2CLK for the AHB2, PCLK for the APB bus peripherals , VPU\_CLK for VPU, DDR\_CLK for DDR.

The Chip has four Phase Locked Loops (PLL):

1. APLL mainly is used to CPU and L2CACHE, also may use the AHB0 and AHB2, other peripherals
2. MPLL mainly is used to DDR, also uses AHB0 and AHB2 APB, other peripherals
3. VPLL mainly is used to VIDEO and other special clocks
4. EPLL mainly is used to AUDIO and other special clocks

**It is recommend DDR clock to use MPLL .**

For the power control logic, there are various power management schemes to keep optimal power consumption for a given task. The power management block can activate four modes: NORMAL mode, DOZE mode, IDLE mode, SLEEP mode.

Support power supply shut down for 4 power domain separately. Software may separately shut down VPU, GPU, GPS, P1 module. When in Sleep mode, software may shut down J1. Thus, the chip may best reduce leakage current.

For reset control logic, the hardware reset and hibernate reset is extended to more 80ms. It controls or distributes all of the system reset signals.

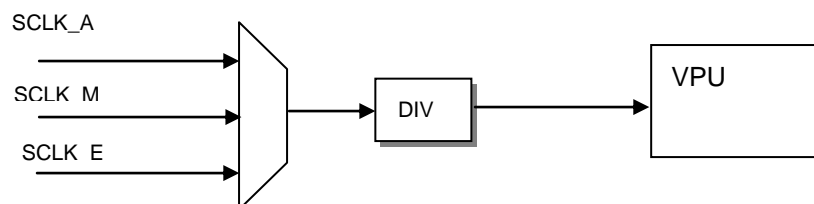
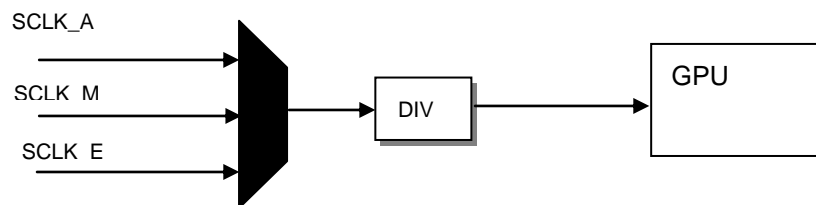
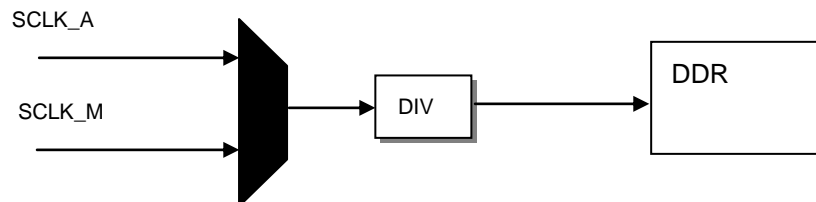
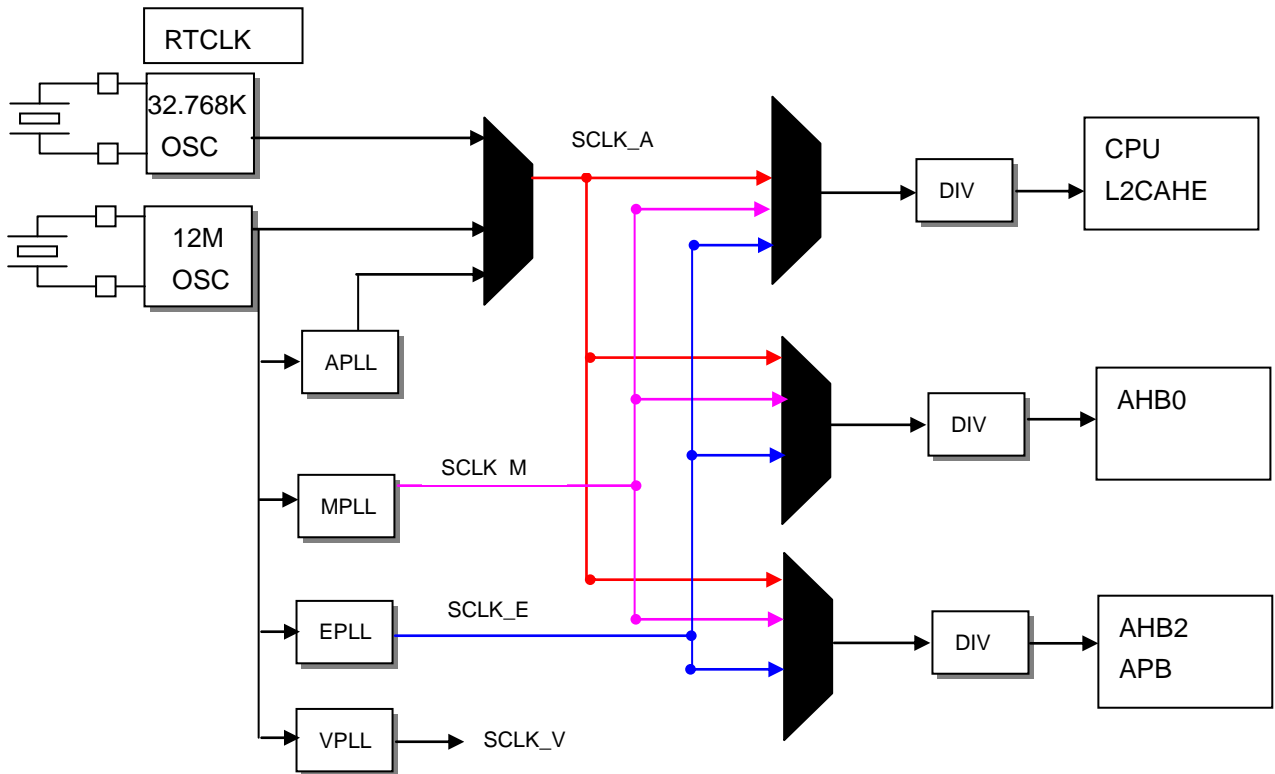
Feature:

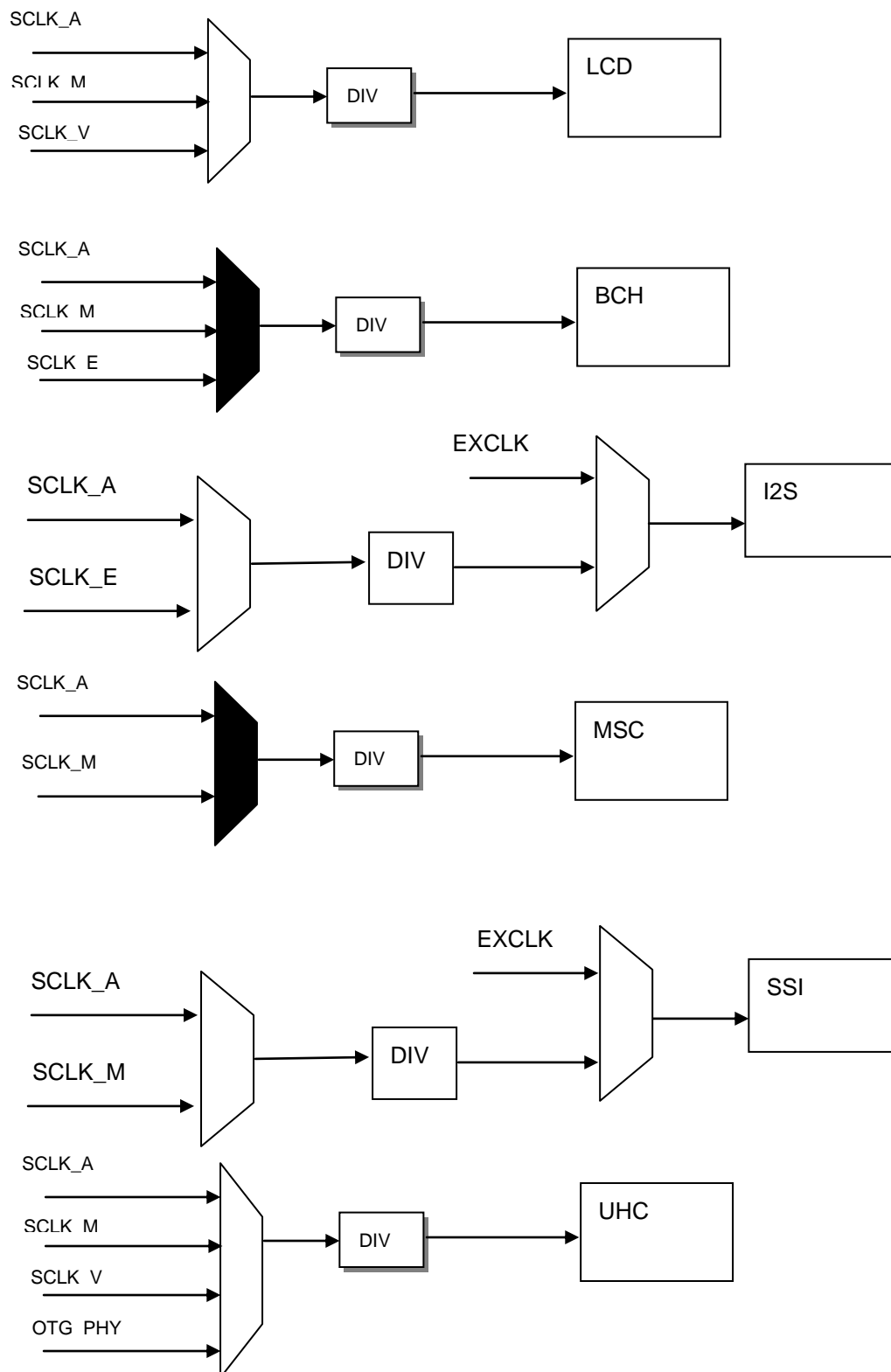
- On-chip 2MHz~27MHz oscillator circuit
- On-chip 32.768KHz oscillator circuit
- One four-chip phase-locked loops (PLL) with programmable multiplier
- CCLK, HHCLK, H2CLK, PCLK, H0CLK, DDR\_CLK, VPU\_CLK frequency can be changed separately for software by setting registers
- SSI clock supports 50M clock
- MSC clock supports 100M clock
- Functional-unit clock gating
- Shut down power supply for J1, VPU, GPU, GPS, P1

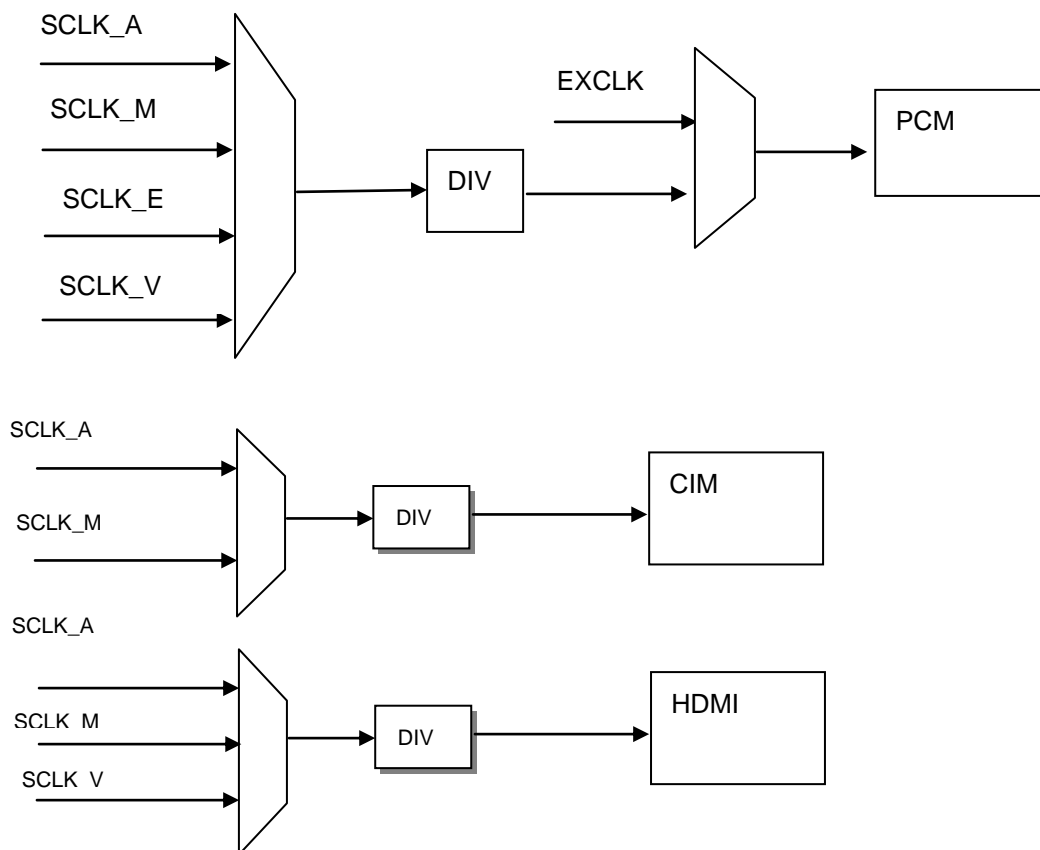


### 18.1.1 CGU Block Diagram

Following figure illustrates a block diagram of CGU.







Clock mux in **grey color** represents **glitch-free** clock mux, which is free of glitches if clock selection is changed. Clock mux in white color represents non-glitch-free clock mux, which can suffer from glitches when changing clock sources. **Care must be taken in using each of clock muxes.** For glitch-free mux, it should be guaranteed that both of clock sources are running when clock selection is changed from one to the other. If that's not the case, clock changing is not finished fully and resulting clock output can have unknown states. For non-glitch-free clock mux, it is possible to have a glitch when clock selections are changed. To prevent the glitch signals, it is recommended to disable output of non-glitch-free muxes before trying to change clock sources. After clock changing is completed, users can re-enable output of the non-glitch-free clock mux so that there will be no glitches resulting from clock changes. Masking output of non-glitch-free muxes are handled by clock source control registers.

### 18.1.2 CGU Registers

All CGU register 32bit access address is physical address.

**Table 18-1 CGU Registers Configuration**

| Name   | description                | RW | Reset Value | Address    | Access Size |
|--------|----------------------------|----|-------------|------------|-------------|
| CPCCR  | Clock Control Register     | RW | 0x95000000  | 0x10000000 | 32          |
| CPCSR  | Clock status register      | RW | 0x00000000  | 0x100000D4 | 32          |
| DDCDR  | DDR clock divider Register | RW | 0x40000000  | 0x1000002C | 32          |
| VPUCDR | VPU clock divider Register | RW | 0x00000000  | 0x10000030 | 32          |

|               |                                    |    |            |            |    |
|---------------|------------------------------------|----|------------|------------|----|
| I2SCDR        | I2S device clock divider Register  | RW | 0x00000000 | 0x10000060 | 32 |
| I2S1CDR       | I2S1 device clock divider Register | RW | 0x00000000 | 0x100000A0 | 32 |
| USBCDR        | OTG PHY clock divider Register     | RW | 0x00000000 | 0x10000050 | 32 |
| LP0CDR        | LCD0 pix clock divider Register    | RW | 0x00000000 | 0x10000054 | 32 |
| LP1CDR        | LCD1 pix clock divider Register    | RW | 0x00000000 | 0x10000064 | 32 |
| MSC0CDR       | MSC0 clock divider Register        | RW | 0x40000000 | 0x10000068 | 32 |
| MSC1CDR       | MSC1 clock divider Register        | RW | 0x00000000 | 0x100000A4 | 32 |
| MSC2CDR       | MSC2 clock divider Register        | RW | 0x00000000 | 0x100000A8 | 32 |
| UHCCDR        | UHC 48M clock divider Register     | RW | 0x00000000 | 0x1000006C | 32 |
| SSICDR        | SSI clock divider Register         | RW | 0x00000000 | 0x10000074 | 32 |
| CIMCDR        | CIM MCLK clock divider Register    | RW | 0x00000000 | 0x1000007C | 32 |
| PCMCDR        | PCM device clock divider Register  | RW | 0x00000000 | 0x10000084 | 32 |
| GPUCDR        | GPU clock divider Register         | RW | 0x40000000 | 0x10000088 | 32 |
| HDMICDR       | HDMI clock divider Register        | RW | 0x00000000 | 0x1000008C | 32 |
| BCHCDR        | BCH clock divider Register         | RW | 0x40000000 | 0x100000AC | 32 |
| CPM_INTR      | CPM interrupt Register             | RW | 0x00000000 | 0x100000B0 | 32 |
| CPM_INTR<br>E | CPM interrupt Enable Register      | RW | 0x00000000 | 0x100000B4 | 32 |
| CPSPR         | CPM Scratch Pad Register           | RW | 0x???????? | 0x10000034 | 32 |
| CPSPPR        | CPM Scratch Protected Register     | RW | 0x0000a5a5 | 0x10000038 | 32 |
| USBPCR        | USB Parameter control register     | RW | 0x429919b8 | 0x1000003C | 32 |
| USBRDT        | USB Reset Detect Timer Register    | RW | 0x02000096 | 0x10000040 | 32 |
| USBVBFIL      | USB jitter filter Register         | RW | 0x00ff0080 | 0x10000044 | 32 |
| USBPCR1       | USB Parameter control register 1   | RW | 0x00000004 | 0x10000048 | 32 |
| CPPCR         | PLL Control Register               | RW | 0x80000020 | 0x1000000C | 32 |
| CPAPCR        | APLL Control Register              | RW | 0x00000000 | 0x10000010 | 32 |
| CPMPCR        | MPLL Control Register              | RW | 0x00000000 | 0x10000014 | 32 |
| CPEPCR        | EPLL Control Register              | RW | 0x00000000 | 0x10000018 | 32 |
| CPVPCR        | VPLL Control Register              | RW | 0x00000000 | 0x1000001C | 32 |

### 18.1.2.1 Clock Control Register

The Clock Control Register (CPCCR) is a 32-bit read/write register, which controls CCLK, H0CLK, H2CLK and PCLK division ratios. It is initialized to 0x95000000 by any reset. Only word access can be used on CPCCR.

| CPCCR |         |    |          |    |           |    |           |    |    |        |         |         |      |    |    | 0x10000000 |    |    |       |    |    |        |   |   |      |   |   |   |   |   |   |   |
|-------|---------|----|----------|----|-----------|----|-----------|----|----|--------|---------|---------|------|----|----|------------|----|----|-------|----|----|--------|---|---|------|---|---|---|---|---|---|---|
| Bit   | 31      | 30 | 29       | 28 | 27        | 26 | 25        | 24 | 23 | 22     | 21      | 20      | 19   | 18 | 17 | 16         | 15 | 14 | 13    | 12 | 11 | 10     | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | SEL_SRC |    | SEL_CPLL |    | SEL_H0PLL |    | SEL_H2PLL |    |    | CE_CPU | CE_AHB2 | CE_AHB2 | PDIV |    |    | H2DIV      |    |    | H0DIV |    |    | L2CDIV |   |   | CDIV |   |   |   |   |   |   |   |
| RST   | 1       | 0  | 0        | 1  | 0         | 1  | 0         | 1  | 0  | 0      | 0       | 0       | 0    | 0  | 0  | 0          | 0  | 0  | 0     | 0  | 0  | 0      | 0 | 0 | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name          | Description                                                                                                                                                                                                                                                                                                   | RW |
|-------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | SEL_SRC       | 00: stop MUX clock output<br>01: APLL<br>10: EXCLK<br>11: RTCLK<br>The mux clock output is SCLK_A<br>The clock mux is free of glitches if clock selection is changed<br>When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running.            | RW |
| 29:28 | SEL_CPLL      | 00: stop MUX clock output<br>01: SCLK_A<br>10: MPLL<br>11: EPLL<br>The mux clock output is to CPU and L2CAHE<br>The clock mux is free of glitches if clock selection is changed<br>When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running. | RW |
| 27:26 | SEL_H0PL<br>L | 00: stop MUX clock output<br>01: SCLK_A<br>10: MPLL<br>11: EPLL<br>The mux clock output is to AHB0<br>The clock mux is free of glitches if clock selection is changed<br>When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running.           | RW |
| 25:24 | SEL_H2PL<br>L | 00 : stop MUX clock output<br>01: SCLK_A<br>10: MPLL<br>11: TCK<br>The mux clock output is to AHB2<br>The clock mux is free of glitches if clock selection is changed<br>When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running.           | RW |
| 23    | Reserved      |                                                                                                                                                                                                                                                                                                               |    |
| 22    | CE_CPU        | Change enable for CPU and L2CAHE. If CE_CPU is 1 , write on CDIV, L2CDIV will start a frequency changing sequence immediately. If CE_CPU is 0, writes on CDIV and L2CDIV have no affect.                                                                                                                      | RW |
| 21    | CE_AHB0       | Change enable for AHB0. If CE_AHB0 is 1 , write on H0DIV will start a frequency changing sequence immediately. If CE_AHB0 is 0, writes on H0DIV have no affect.                                                                                                                                               | RW |
| 20    | CE_AHB2       | Change enable for AHB2. If CE_AHB2 is 1 , write on H2DIV, PDIV will start a frequency changing sequence immediately. If CE_AHB2 is 0, writes on H2DIV, PDIV have no affect.                                                                                                                                   | RW |

| 19:16            | PDIV  | <p>Divider for Peripheral Frequency. Specified the PCLK division ratio.</p> <table border="1" data-bbox="523 271 1302 1084"> <thead> <tr> <th colspan="4">Bit 19~16: PDIV</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>X1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>X1/2</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>X1/3</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>X1/4</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>X1/5</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>X1/6</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>X1/7</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>X1/8</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>X1/9</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>X1/10</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>X1/11</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>X1/12</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>X1/13</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>X1/14</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td><b>Clr stop<br/>CLK</b></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td><b>Set Stop<br/>CLK</b></td></tr> </tbody> </table> <p>When software sets PDIV == 1111 and CE is 1, PCLK will stop. PDIV value will not be changed actually, it is only set register.</p> <p>When software sets PDIV == 1110 and CE is 1, PCLK will continue to run. PDIV value will not be changed actually, it is only clear register.</p> | Bit 19~16: PDIV  |                         |  |  | Description | 0 | 0 | 0 | 0 | X1 | 0 | 0 | 0 | 1 | X1/2 | 0 | 0 | 1 | 0 | X1/3 | 0 | 0 | 1 | 1 | X1/4 | 0 | 1 | 0 | 0 | X1/5 | 0 | 1 | 0 | 1 | X1/6 | 0 | 1 | 1 | 0 | X1/7 | 0 | 1 | 1 | 1 | X1/8 | 1 | 0 | 0 | 0 | X1/9 | 1 | 0 | 0 | 1 | X1/10 | 1 | 0 | 1 | 0 | X1/11 | 1 | 0 | 1 | 1 | X1/12 | 1 | 1 | 0 | 0 | X1/13 | 1 | 1 | 0 | 1 | X1/14 | 1 | 1 | 1 | 0 | <b>Clr stop<br/>CLK</b> | 1  | 1 | 1 | 1 | <b>Set Stop<br/>CLK</b> | RW |
|------------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-------------------------|--|--|-------------|---|---|---|---|----|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------------------------|----|---|---|---|-------------------------|----|
| Bit 19~16: PDIV  |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                  | Description             |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 0     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1                      |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 0     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/2                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 0     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/3                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 0     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/4                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 1     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/5                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 1     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/6                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 1     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/7                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 1     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/8                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 0     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/9                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 0     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/10                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 0     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/11                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 0     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/12                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 1     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/13                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 1     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/14                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 1     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | <b>Clr stop<br/>CLK</b> |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 1     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | <b>Set Stop<br/>CLK</b> |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 15:12            | H2DIV | <p>Divider for AHB2 Frequency. Specified the AHB2 CLK division ratio.</p> <table border="1" data-bbox="523 1339 1302 2018"> <thead> <tr> <th colspan="4">Bit 15~12: H2DIV</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>X1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>X1/2</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>X1/3</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>X1/4</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>X1/5</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>X1/6</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>X1/7</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>X1/8</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>X1/9</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>X1/10</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>X1/11</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>X1/12</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>X1/13</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>X1/14</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td><b>Clr stop</b></td></tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                    | Bit 15~12: H2DIV |                         |  |  | Description | 0 | 0 | 0 | 0 | X1 | 0 | 0 | 0 | 1 | X1/2 | 0 | 0 | 1 | 0 | X1/3 | 0 | 0 | 1 | 1 | X1/4 | 0 | 1 | 0 | 0 | X1/5 | 0 | 1 | 0 | 1 | X1/6 | 0 | 1 | 1 | 0 | X1/7 | 0 | 1 | 1 | 1 | X1/8 | 1 | 0 | 0 | 0 | X1/9 | 1 | 0 | 0 | 1 | X1/10 | 1 | 0 | 1 | 0 | X1/11 | 1 | 0 | 1 | 1 | X1/12 | 1 | 1 | 0 | 0 | X1/13 | 1 | 1 | 0 | 1 | X1/14 | 1 | 1 | 1 | 0 | <b>Clr stop</b>         | RW |   |   |   |                         |    |
| Bit 15~12: H2DIV |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                  | Description             |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 0     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1                      |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 0     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/2                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 0     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/3                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 0     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/4                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 1     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/5                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 1     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/6                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 1     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/7                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 0                | 1     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/8                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 0     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/9                    |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 0     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/10                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 0     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/11                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 0     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/12                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 1     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | X1/13                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 1     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1                | X1/14                   |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |
| 1                | 1     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0                | <b>Clr stop</b>         |  |  |             |   |   |   |   |    |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |    |   |   |   |                         |    |

|                 |          | <table border="1"> <tr> <td></td> <td></td> <td></td> <td></td> <td><b>CLK</b></td> </tr> <tr> <td><b>1</b></td> <td><b>1</b></td> <td><b>1</b></td> <td><b>1</b></td> <td><b>Set Stop CLK</b></td> </tr> </table> <p>When software sets H2DIV == 1111 and CE is 1, H2CLK will stop. H2DIV value will not be changed actually, it is only set register.</p> <p>When software sets H2DIV == 1110 and CE is 1, H2CLK will continue to run. H2DIV value will not be changed actually, it is only clear register.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                 |                     |  |  | <b>CLK</b>  | <b>1</b> | <b>1</b> | <b>1</b> | <b>1</b> | <b>Set Stop CLK</b> |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
|-----------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------------------|--|--|-------------|----------|----------|----------|----------|---------------------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|---|---|---|------|----|---|---|---|------|---|---|---|---|------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|----------|----------|----------|----------|---------------------|----------|----------|----------|----------|---------------------|----|
|                 |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 | <b>CLK</b>          |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| <b>1</b>        | <b>1</b> | <b>1</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <b>1</b>        | <b>Set Stop CLK</b> |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 11:8            | H0DIV    | <p>Divider for AHB0 Frequency. Specified the AHB0 CLK division ratio.</p> <table border="1"> <thead> <tr> <th colspan="4">Bit 11~8: H0DIV</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>X1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>X1/2</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>X1/3</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>X1/4</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>X1/5</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>X1/6</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>X1/7</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>X1/8</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>X1/9</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>X1/10</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>X1/11</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>X1/12</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>X1/13</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>X1/14</td></tr> <tr><td><b>1</b></td><td><b>1</b></td><td><b>1</b></td><td><b>0</b></td><td><b>Clr stop CLK</b></td></tr> <tr><td><b>1</b></td><td><b>1</b></td><td><b>1</b></td><td><b>1</b></td><td><b>Set Stop CLK</b></td></tr> </tbody> </table> <p>When software sets H0DIV == 1111 and CE is 1, H0CLK will stop. H0DIV value will not be changed actually, it is only set register.</p> <p>When software sets H0DIV == 1110 and CE is 1, H0CLK will continue to run. H0DIV value will not be changed actually, it is only clear register.</p> | Bit 11~8: H0DIV |                     |  |  | Description | 0        | 0        | 0        | 0        | X1                  | 0 | 0 | 0 | 1 | X1/2 | 0 | 0 | 1 | 0 | X1/3 | 0 | 0 | 1 | 1 | X1/4 | 0 | 1 | 0 | 0 | X1/5 | 0 | 1 | 0 | 1 | X1/6 | 0 | 1 | 1 | 0 | X1/7 | 0  | 1 | 1 | 1 | X1/8 | 1 | 0 | 0 | 0 | X1/9 | 1 | 0 | 0 | 1 | X1/10 | 1 | 0 | 1 | 0 | X1/11 | 1 | 0 | 1 | 1 | X1/12 | 1 | 1 | 0 | 0 | X1/13 | 1 | 1 | 0 | 1 | X1/14 | <b>1</b> | <b>1</b> | <b>1</b> | <b>0</b> | <b>Clr stop CLK</b> | <b>1</b> | <b>1</b> | <b>1</b> | <b>1</b> | <b>Set Stop CLK</b> | RW |
| Bit 11~8: H0DIV |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 | Description         |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 0        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1                  |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 0        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/2                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 0        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1/3                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 0        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/4                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 1        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1/5                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 1        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/6                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 1        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1/7                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 1        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/8                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 1               | 0        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1/9                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 1               | 0        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/10               |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 1               | 0        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1/11               |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 1               | 0        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/12               |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 1               | 1        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1/13               |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 1               | 1        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/14               |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| <b>1</b>        | <b>1</b> | <b>1</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <b>0</b>        | <b>Clr stop CLK</b> |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| <b>1</b>        | <b>1</b> | <b>1</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <b>1</b>        | <b>Set Stop CLK</b> |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 7:4             | L2CDIV   | <p>Divider for L2CACHE Frequency. Specified the L2C CLK division ratio.</p> <table border="1"> <thead> <tr> <th colspan="4">Bit 7~4: L2CDIV</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>X1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>X1/2</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>X1/3</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>X1/4</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>X1/5</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>X1/6</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>X1/7</td></tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Bit 7~4: L2CDIV |                     |  |  | Description | 0        | 0        | 0        | 0        | X1                  | 0 | 0 | 0 | 1 | X1/2 | 0 | 0 | 1 | 0 | X1/3 | 0 | 0 | 1 | 1 | X1/4 | 0 | 1 | 0 | 0 | X1/5 | 0 | 1 | 0 | 1 | X1/6 | 0 | 1 | 1 | 0 | X1/7 | RW |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| Bit 7~4: L2CDIV |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 | Description         |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 0        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1                  |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 0        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/2                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 0        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1/3                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 0        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/4                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 1        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1/5                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 1        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1               | X1/6                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |
| 0               | 1        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0               | X1/7                |  |  |             |          |          |          |          |                     |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |   |   |   |   |      |    |   |   |   |      |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |          |          |          |          |                     |          |          |          |          |                     |    |

|               |      | <table border="1"> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>X1/8</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>X1/9</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>X1/10</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>X1/11</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>X1/12</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>X1/13</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>X1/14</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td><b>Clr stop<br/>CLK</b></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td><b>Set Stop<br/>CLK</b></td></tr> </table> <p>When software sets L2CDIV == 1111 and CE is 1, L2CCLK will stop. L2CDIV value will not be changed actually, it is only set register.</p> <p>When software sets L2CDIV == 1110 and CE is 1, L2CCLK will continue to run. L2CDIV value will not be changed actually, it is only clear register.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0             | 1                       | 1 | 1 | X1/8        | 1 | 0 | 0 | 0 | X1/9 | 1 | 0 | 0 | 1 | X1/10 | 1 | 0 | 1 | 0 | X1/11 | 1 | 0 | 1 | 1 | X1/12 | 1 | 1 | 0 | 0 | X1/13 | 1 | 1 | 0 | 1 | X1/14 | 1 | 1 | 1 | 0 | <b>Clr stop<br/>CLK</b> | 1 | 1 | 1 | 1 | <b>Set Stop<br/>CLK</b> |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
|---------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-------------------------|---|---|-------------|---|---|---|---|------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------------------------|---|---|---|---|-------------------------|---|---|---|---|------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------|---|---|---|---|-------------------------|---|---|---|---|-------------------------|
| 0             | 1    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/8                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 0    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1/9                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 0    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/10                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 0    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1/11                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 0    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/12                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 1    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1/13                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 1    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/14                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 1    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | <b>Clr stop<br/>CLK</b> |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 1    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | <b>Set Stop<br/>CLK</b> |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 3:0           | CDIV | <p>Divider for CPU Frequency. Specified the CPU CLK division ratio.</p> <table border="1"> <thead> <tr> <th colspan="4">Bit 3~0: CDIV</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>X1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>X1/2</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>X1/3</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>X1/4</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>X1/5</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>X1/6</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>X1/7</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>X1/8</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>X1/9</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>X1/10</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>X1/11</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>X1/12</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>X1/13</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>X1/14</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td><b>Clr stop<br/>CLK</b></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td><b>Set Stop<br/>CLK</b></td></tr> </tbody> </table> <p>When software sets CDIV == 1111 and CE is 1, CCLK will stop. CDIV value will not be changed actually, it is only set register.</p> <p>When software sets CDIV == 1110 and CE is 1, CCLK will continue to run. CDIV value will not be changed actually, it is only clear register.</p> | Bit 3~0: CDIV |                         |   |   | Description | 0 | 0 | 0 | 0 | X1   | 0 | 0 | 0 | 1 | X1/2  | 0 | 0 | 1 | 0 | X1/3  | 0 | 0 | 1 | 1 | X1/4  | 0 | 1 | 0 | 0 | X1/5  | 0 | 1 | 0 | 1 | X1/6  | 0 | 1 | 1 | 0 | X1/7                    | 0 | 1 | 1 | 1 | X1/8                    | 1 | 0 | 0 | 0 | X1/9 | 1 | 0 | 0 | 1 | X1/10 | 1 | 0 | 1 | 0 | X1/11 | 1 | 0 | 1 | 1 | X1/12 | 1 | 1 | 0 | 0 | X1/13 | 1 | 1 | 0 | 1 | X1/14 | 1 | 1 | 1 | 0 | <b>Clr stop<br/>CLK</b> | 1 | 1 | 1 | 1 | <b>Set Stop<br/>CLK</b> |
| Bit 3~0: CDIV |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |               | Description             |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 0             | 0    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1                      |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 0             | 0    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/2                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 0             | 0    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1/3                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 0             | 0    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/4                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 0             | 1    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1/5                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 0             | 1    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/6                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 0             | 1    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1/7                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 0             | 1    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/8                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 0    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1/9                    |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 0    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/10                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 0    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1/11                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 0    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/12                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 1    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | X1/13                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 1    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | X1/14                   |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 1    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0             | <b>Clr stop<br/>CLK</b> |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |
| 1             | 1    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1             | <b>Set Stop<br/>CLK</b> |   |   |             |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |   |   |   |   |      |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |       |   |   |   |   |                         |   |   |   |   |                         |



**Import Note :** for PDIV and AHB2DIV, AHB2's clk frequency must be 1 or 2 times of PDIV's clk frequency. For L2CDIV and CDIV, CPU's clk frequency must be 1,2,3, or 4 times of L2CACHE's clk frequency.

### 18.1.2.2 Clock Status Register

Clock status register (CPCSR) is a 32-bit read/write register that specifies the status of CPCCR .This register is initialized to 0x00000000 only by any reset. Only word access can be used on CPCSR.

| CPCSR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x100000D4 |    |    |    |    |    |    |   |   |   |   |   |   |            |            |           |   |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|------------|------------|-----------|---|
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3          | 2          | 1         | 0 |
|       | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   | H2DIV_BUSY | H0DIV_BUSY | CDIV_BUSY |   |
| RST   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0          | 0          | 0         | 0 |

| Bits | Name           | Description                                                                                                                                                                                                                                                                                           | RW |
|------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:3 | Reserved       | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                  | R  |
| 2    | H2DIV_B<br>USY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until H2DIV_BUSY == 0, then may begin another frequency change. | R  |
| 1    | H0DIV_B<br>USY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until H0DIV_BUSY == 0, then may begin another frequency change. | R  |
| 0    | CDIV_B<br>USY  | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until CDIV_BUSY == 0, then may begin another frequency change.  | R  |

### 18.1.2.3 DDR Memory clock divider Register

DDR memory clock divider Register (DDRCDR) is a 32-bit read/write register that specifies the divider of DDR memory clock . This register is initialized to 0x40000000 only by any reset. Only word access can be used on DDRCDR.

|        |     |    |    |    |        |          |          |          |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |        |   |   |   |   |   |   |
|--------|-----|----|----|----|--------|----------|----------|----------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|--------|---|---|---|---|---|---|
| DDRCDR |     |    |    |    |        |          |          |          |    |    |    |    |    |    |    | 0x1000002C |    |    |    |    |    |    |   |   |   |   |        |   |   |   |   |   |   |
| Bit    | 31  | 30 | 29 | 28 | 27     | 26       | 25       | 24       | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5      | 4 | 3 | 2 | 1 | 0 |   |
|        | DCS |    |    |    | CE_DDR | DDR_BUSY | DDR_STOP | Reserved |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   | DDRCDR |   |   |   |   |   |   |
| RST    | 0   | 1  | 0  | 0  | 0      | 0        | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                   | RW |
|-------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | DCS      | DDR Clock Source Selection. Selects the DDR clock source between APLL output and MPLL output.<br>00: stop MUX clock output<br>01: DDR clock source is SCLK_A<br>10: DDR clock source is MPLL<br>The clock mux is free of glitches if clock selection is changed<br>When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running. | RW |
| 29    | CE_DDR   | Change enable for DDR. If CE_DDR is 1 , write on DDRCDR will start a frequency changing sequence immediately. If CE_DDR is 0, writes on DDRCDR have no affect                                                                                                                                                                                                                                 | RW |
| 28    | DDR_BUSY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until DDR_BUSY == 0, then may begin another frequency change.                                                                                           | RW |
| 27    | DDR_STOP | When DDR_STOP is 1 and CE_DDR is 1, the DDR clock will stop.<br>When DDR_STOP is 0 and CE_DDR is 1, the DDR clock will continue.<br>Software should wait until DDR_BUSY == 0, then may begin another frequency change.<br>DDR_STOP is prior to DDRCDR.                                                                                                                                        | RW |
| 26:4  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                          | R  |
| 3:0   | DDRCDR   | Divider for DDR Frequency. Specified the DDR memory clock division ratio, which varies from 1 to 16 (division ratio = DDRCDR + 1).                                                                                                                                                                                                                                                            | RW |

### 18.1.2.4 VPU clock divider Register

VPU clock divider Register (VPUCDR) is a 32-bit read/write register that specifies the divider of VPU clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on VPUCDR.

| VPUCDR |     |    |    |    |        |          |          |          |    |    |    |    |    |    | 0x10000030 |    |    |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
|--------|-----|----|----|----|--------|----------|----------|----------|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|---|---|--------|---|---|---|---|---|---|---|
| Bit    | 31  | 30 | 29 | 28 | 27     | 26       | 25       | 24       | 23 | 22 | 21 | 20 | 19 | 18 | 17         | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | VCS |    |    |    | CE_VPU | VPU_BUSY | VPU_STOP | Reserved |    |    |    |    |    |    |            |    |    |    |    |    |    |    |   |   | VPUCDR |   |   |   |   |   |   |   |
| RST    | 0   | 0  | 0  | 0  | 0      | 0        | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                              | RW |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | VCS      | VPU Clock Source Selection. Selects the VPU clock source<br>00: VPU clock source is SCLK_A<br>01: VPU clock source is MPLL<br>10: VPU clock source is EPLL<br>It is no glitch free mux. Software should be stop VPU clock, when change this bit.                                                                         | RW |
| 29    | CE_VPU   | Change enable for VPU. If CE_VPU is 1 , write on VPUCDR will start a frequency changing sequence immediately. If CE_VPU is 0, writes on VPUCDR have no affect                                                                                                                                                            | RW |
| 28    | VPU_BUSY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until VPU_BUSY == 0, may begin another frequency change.                           | RW |
| 27    | VPU_STOP | When VPU_STOP is 1 and CE_VPU is 1, the VPU clock will stop.<br>When VPU_STOP is 0 and CE_VPU is 1, the VPU clock will continue.<br>Software should wait until VPU_BUSY == 0, may begin another frequency change.<br>VPU_STOP is prior to VPUCDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 26:4  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                     | R  |
| 3:0   | VPUCDR   | Divider for VPU Frequency. Specified the VPU clock division ratio, which varies from 1 to 16 (division ratio = VPUCDR + 1).                                                                                                                                                                                              | RW |

### 18.1.2.5 I2S device clock divider Register//aic

I2S device clock divider Register (I2SCDR) is a 32-bit read/write register that specifies the divider of I2S device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on I2SCDR.

|            |               |       |        |          |          |          |    |    |    |    |    |    |    |    |    |    |                  |    |    |    |    |        |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
|------------|---------------|-------|--------|----------|----------|----------|----|----|----|----|----|----|----|----|----|----|------------------|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|
|            | <b>I2SCDR</b> |       |        |          |          |          |    |    |    |    |    |    |    |    |    |    | <b>0x1000060</b> |    |    |    |    |        |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
| <b>Bit</b> | 31            | 30    | 29     | 28       | 27       | 26       | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15               | 14 | 13 | 12 | 11 | 10     | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|            | I2CS          | I2PCS | CE_I2S | I2S_BUSY | I2S_STOP | Reserved |    |    |    |    |    |    |    |    |    |    |                  |    |    |    |    | I2SCDR |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
| <b>RST</b> | 0             | 0     | 0      | 0        | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                | 0  | 0  | 0  | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                           | RW |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31   | I2CS     | I2S Clock Source Selection. Selects the I2S clock source between PLL output and pin EXCLK.<br>0: I2S clock source is EXCLK<br>1: I2S clock source is PLL output divided by I2SDIV<br>It is no glitch free mux. Software should be stop I2S clock, when change this bit.<br>If I2CS == 0, the clock divider is gated, don't set CE_I2S | RW |
| 30   | I2PCS    | 0: SCLK_A output<br>1: EPLL output                                                                                                                                                                                                                                                                                                    | RW |
| 29   | CE_I2S   | Change enable for I2S. If CE_I2S is 1 , write on I2SCDR ill start a frequency changing sequence immediately. If CE_I2S is 0, writes on I2SCDR have no affect                                                                                                                                                                          | RW |
| 28   | I2S_BUSY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until I2S_BUSY == 0, may begin another frequency change.                                        | R  |
| 27   | I2S_STOP | When I2S_STOP is 1 and CE_I2S is 1, the I2S clock will stop.<br>When I2S_STOP is 0 and CE_I2S is 1, the I2S clock will continue.<br>Software should wait until I2S_BUSY == 0, may begin another frequency change.<br>I2S_STOP is prior to I2SCDR.<br>If system don't use this module, may set this bit to reduce power .              | RW |
| 26:8 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                  |    |
| 7:0  | I2SCDR   | Divider for I2S Frequency. Specified the I2S device clock division ratio, which varies from 1 to 256 (division ratio = I2SCDR + 1).                                                                                                                                                                                                   | RW |

### 18.1.2.6 LCD0 pix clock divider Register

LCD pix clock divider Register (LPCDR) is a 32-bit read/write register that specifies the divider of LCD pixel clock (LPCLK). This register is initialized to 0x04000000 only by any reset. Only word access can be used on LPCDR.

| LPCDR |      |    |    |    |    |      |    |        |    |          |    |          |    |          |    | 0x10000054 |    |    |    |    |    |    |   |   |   |   |   |   |   |       |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |
|-------|------|----|----|----|----|------|----|--------|----|----------|----|----------|----|----------|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|-------|---|---|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Bit   | 31   | 30 | 29 | 28 | 27 | 26   | 25 | 24     | 23 | 22       | 21 | 20       | 19 | 18       | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2     | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |
|       | LPCS |    |    |    |    | LTCS |    | CE_LCD |    | LCD_BUSY |    | LCD_STOP |    | Reserved |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   | LPCDR |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RST   | 0    | 0  | 0  | 0  | 0  | 0    | 0  | 0      | 0  | 0        | 0  | 0        | 0  | 0        | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                             | RW |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | LPCS     | 00: SCLK_A output<br>01: MPLL output<br>10: VPLL output<br>It is no glitch free mux. Software should be stop PIX clock, when change this bit.                                                                                                                                                                           | RW |
| 29    | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                    | RW |
| 28    | CE_LCD   | Change enable for LCD. If CE_LCD is 1 , write on LPCDR will start a frequency changing sequence immediately. If CE_LCD is 0, writes on LPCDR have no affect                                                                                                                                                             | RW |
| 27    | LCD_BUSY | The bit is read only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until LCD_BUSY == 0, may begin another frequency change.                           | RW |
| 26    | LCD_STOP | When LCD_STOP is 1 and CE_LCD is 1, the LCD clock will stop.<br>When LCD_STOP is 0 and CE_LCD is 1, the LCD clock will continue.<br>Software should wait until LCD_BUSY == 0, may begin another frequency change.<br>LCD_STOP is prior to LPCDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 25:8  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                    | RW |
| 7:0   | LPCDR    | Divider for Pixel Frequency. Specified the LCD pixel clock (LPCLK) division ratio, which varies from 1 to 256 (division ratio = LPCDR + 1).                                                                                                                                                                             | RW |

### 18.1.2.7 LCD1 pix clock divider Register

LCD1 pix clock divider Register (LPCDR1) is a 32-bit read/write register that specifies the divider of LCD1 pixel clock (LPCLK). This register is initialized to 0x04000000 only by any reset. Only word access can be used on LPCDR.

|        |      |    |      |        |          |          |          |    |    |    |    |    |    |    |    |            |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|------|----|------|--------|----------|----------|----------|----|----|----|----|----|----|----|----|------------|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| LP1CDR |      |    |      |        |          |          |          |    |    |    |    |    |    |    |    | 0x10000064 |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit    | 31   | 30 | 29   | 28     | 27       | 26       | 25       | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | LPCS |    | LTC5 | CE_LCD | LCD_BUSY | LCD_STOP | Reserved |    |    |    |    |    |    |    |    |            | LPCDR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST    | 0    | 0  | 0    | 0      | 0        | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0     | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name          | Description                                                                                                                                                                                                                                                                                                             | RW |
|-------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | LP1CS         | 00: SCLK_A output<br>01: MPLL output<br>10: VPLL output<br>It is no glitch free mux. Software should be stop PIX clock, when change this bit.                                                                                                                                                                           | RW |
| 29    | Reserved      | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                    | RW |
| 28    | CE_LCD<br>1   | Change enable for LCD. If CE_LCD is 1 , write on LPCDR will start a frequency changing sequence immediately. If CE_LCD is 0, writes on LPCDR have no affect                                                                                                                                                             | RW |
| 27    | LCD1_B<br>USY | The bit is read only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until LCD_BUSY == 0, may begin another frequency change.                           | RW |
| 26    | LCD1_S<br>TOP | When LCD_STOP is 1 and CE_LCD is 1, the LCD clock will stop.<br>When LCD_STOP is 0 and CE_LCD is 1, the LCD clock will continue.<br>Software should wait until LCD_BUSY == 0, may begin another frequency change.<br>LCD_STOP is prior to LPCDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 25:8  | Reserved      | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                    | RW |
| 7:0   | LP1CDR        | Divider for Pixel Frequency. Specified the LCD pixel clock (LPCLK) division ratio, which varies from 1 to 256 (division ratio = LPCDR + 1).                                                                                                                                                                             | RW |

### 18.1.2.8 MSC0 device clock divider Register

MSC0 device clock divider Register (MSC0CDR) is a 32-bit read/write register that specifies the divider of MSC0 device clock . This register is initialized to 0x40000000 only by any reset. Only word access can be used on MSC0CDR.

| MSC0CDR |      |    |    |    |         |           |           |          |    |    |    |    |    |    |    | 0x10000068 |            |          |    |    |    |    |   |         |   |   |   |   |   |   |   |   |
|---------|------|----|----|----|---------|-----------|-----------|----------|----|----|----|----|----|----|----|------------|------------|----------|----|----|----|----|---|---------|---|---|---|---|---|---|---|---|
| Bit     | 31   | 30 | 29 | 28 | 27      | 26        | 25        | 24       | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15         | 14       | 13 | 12 | 11 | 10 | 9 | 8       | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|         | MPCS |    |    |    | CE_MSC0 | MSC0_BUSY | MSC0_STOP | Reserved |    |    |    |    |    |    |    |            | S_CLK0_SEL | Reserved |    |    |    |    |   | MSC0CDR |   |   |   |   |   |   |   |   |
| RST     | 0    | 1  | 0  | 0  | 0       | 0         | 0         | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0          | 0        | 0  | 0  | 0  | 0  | 0 | 0       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name       | Description                                                                                                                                                                                                                                                                                                                   | RW |
|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | MPCS       | 00: stop mux clock output<br>01: select SCLK_A clock output<br>10: select MPLL clock output<br>The clock mux is free of glitches if clock selection is changed<br>The clock mux output is to MSC0, MSC1, MSC2 . It is source clock of MSC0, MSC1, MSC2.                                                                       | RW |
| 29    | CE_MSC0    | Change enable for MSC0. If CE_MSC0 is 1 , write on MSC0CDR ill start a frequency changing sequence immediatly. If CE_MSC0 is 0, writes on MSC0CDR have no affect                                                                                                                                                              | RW |
| 28    | MSC0_BUSY  | The bit is read only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until MSC0_BUSY == 0, may begin another frequency change.                                | RW |
| 27    | MSC0_STOP  | When MSC0_STOP is 1 and CE_MSC0 is 1, the MSC0 clock will stop.<br>When MSC0_STOP is 0 and CE_MSC0 is 1, the MSC0 clock will continue.<br>Software should wait until MSC0_BUSY == 0, may begin another frequency change.MSC0_STOP is prior to MSC0CDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 26:16 | Reserved   | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                          | R  |
| 15    | S_CLK0_SEL | MSC sample clock selection:<br>0: Sample clock is 90-degree phase shifted by device clock<br>1: Sample clock is 180-degree phase shifted by device clock                                                                                                                                                                      | RW |
| 14:8  | Reserved   |                                                                                                                                                                                                                                                                                                                               |    |
| 7:0   | MSC0CDR    | Divider for MSC0 Frequency. Specified the MSC0 device clock division ratio division ratio = (MSC0CDR + 1)*2.                                                                                                                                                                                                                  | RW |

### 18.1.2.9 MSC1 device clock divider Register

MSC1 device clock divider Register (MSC1CDR) is a 32-bit read/write register that specifies the divider of MSC1 device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on MSC1CDR.

|                |          |    |         |           |           |          |    |    |    |    |    |    |    |    |    |                   |          |    |    |    |    |    |         |   |   |   |   |   |   |   |   |   |
|----------------|----------|----|---------|-----------|-----------|----------|----|----|----|----|----|----|----|----|----|-------------------|----------|----|----|----|----|----|---------|---|---|---|---|---|---|---|---|---|
| <b>MSC1CDR</b> |          |    |         |           |           |          |    |    |    |    |    |    |    |    |    | <b>0x100000A4</b> |          |    |    |    |    |    |         |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>     | 31       | 30 | 29      | 28        | 27        | 26       | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15       | 14 | 13 | 12 | 11 | 10 | 9       | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|                | Reserved |    | CE_MSC1 | MSC1_BUSY | MSC1_STOP | Reserved |    |    |    |    |    |    |    |    |    | S_CLK1_SEL        | Reserved |    |    |    |    |    | MSC1CDR |   |   |   |   |   |   |   |   |   |
| <b>RST</b>     | 0        | 0  | 0       | 0         | 0         | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0        | 0  | 0  | 0  | 0  | 0  | 0       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name       | Description                                                                                                                                                                                                                                                                                                                   | RW |
|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | Reserved   |                                                                                                                                                                                                                                                                                                                               | R  |
| 29    | CE_MSC1    | Change enable for MSC1. If CE_MSC1 is 1 , write on MSC1CDR will start a frequency changing sequence immediately. If CE_MSC1 is 0, writes on MSC1CDR have no affect                                                                                                                                                            | RW |
| 28    | MSC1_BUSY  | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until MSC1_BUSY == 0, may go next                                                       | RW |
| 27    | MSC1_STOP  | When MSC1_STOP is 1 and CE_MSC1 is 1, the MSC1 clock will stop.<br>When MSC1_STOP is 0 and CE_MSC1 is 1, the MSC1 clock will continue.<br>Software should wait until MSC1_BUSY == 0, may begin another frequency change.MSC1_STOP is prior to MSC1CDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 26:16 | Reserved   | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                          | R  |
| 15    | S_CLK1_SEL | MSC1 sample clock selection:<br>0: Sample clock is 90-degree phase shifted by device clock<br>1: Sample clock is 180-degree phase shifted by device clock                                                                                                                                                                     | RW |
| 14:8  | Reserved   |                                                                                                                                                                                                                                                                                                                               |    |
| 7:0   | MSC1CDR    | Divider for MSC1 Frequency. Specified the MSC1 device clock division ratio division ratio = (MSC1CDR + 1)*2.                                                                                                                                                                                                                  | RW |

### 18.1.2.10 MSC2 device clock divider Register

MSC2 device clock divider Register (MSC2CDR) is a 32-bit read/write register that specifies the divider of MSC2 device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on MSC2CDR.

|                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>MSC2CDR</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x100000A8</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>     | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |



|     |          |   |         |           |           |          |   |   |   |   |   |   |   |   |   |            |          |   |   |   |   |         |   |   |   |   |   |   |   |   |   |   |   |
|-----|----------|---|---------|-----------|-----------|----------|---|---|---|---|---|---|---|---|---|------------|----------|---|---|---|---|---------|---|---|---|---|---|---|---|---|---|---|---|
|     | Reserved |   | CE_MSC2 | MSC2_BUSY | MSC2_STOP | Reserved |   |   |   |   |   |   |   |   |   | S_CLK2_SEL | Reserved |   |   |   |   | MSC2CDR |   |   |   |   |   |   |   |   |   |   |   |
| RST | 0        | 0 | 0       | 0         | 0         | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0          | 0        | 0 | 0 | 0 | 0 | 0       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name       | Description                                                                                                                                                                                                                                                                                                                   | RW |
|-------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | Reserved   |                                                                                                                                                                                                                                                                                                                               | R  |
| 29    | CE_MSC2    | Change enable for MSC2. If CE_MSC2 is 1 , write on MSC2CDR will start a frequency changing sequence immediately. If CE_MSC2 is 0, writes on MSC2CDR have no affect                                                                                                                                                            | RW |
| 28    | MSC2_BUSY  | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until MSC2_BUSY == 0, may begin another frequency change.                               | RW |
| 27    | MSC2_STOP  | When MSC2_STOP is 1 and CE_MSC2 is 1, the MSC2 clock will stop.<br>When MSC2_STOP is 0 and CE_MSC2 is 1, the MSC2 clock will continue.<br>Software should wait until MSC2_BUSY == 0, may begin another frequency change.MSC2_STOP is prior to MSC2CDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 26:16 | Reserved   | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                          | R  |
| 15    | S_CLK2_SEL | MSC2 sample clock selection:<br>0: Sample clock is 90-degree phase shifted by device clock<br>1: Sample clock is 180-degree phase shifted by device clock                                                                                                                                                                     | RW |
| 14:8  | Reserved   |                                                                                                                                                                                                                                                                                                                               |    |
| 7:0   | MSC2CDR    | Divider for MSC2 Frequency. Specified the MSC2 device clock division ratio division ratio = (MSC2CDR + 1)*2.                                                                                                                                                                                                                  | RW |

### 18.1.2.11 UHC device clock divider Register

UHC device clock divider Register (UHCCDR) is a 32-bit read/write register that specifies the divider of UHC 48M device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on UHCCDR.

|        |      |    |        |          |          |          |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|------|----|--------|----------|----------|----------|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| UHCCDR |      |    |        |          |          |          |    |    |    |    |    |    |    |    |    | 0x1000006C |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit    | 31   | 30 | 29     | 28       | 27       | 26       | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | UHCs |    | CE_UHC | UHC_BUSY | UHC_STOP | Reserved |    |    |    |    |    |    |    |    |    | UHCCDR     |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST    | 0    | 0  | 0      | 0        | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits      | Name     | Description                                                                                                                                                                                                                                                                                                              | RW |
|-----------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:3<br>0 | UHCS     | 00: UHC clock source is SCLK_A output<br>01: UHC clock source is MPLL output<br>10: UHC clock source is EPLL output<br>11: UHC clock source is OTG_PHY<br>It is no glitch free mux. Software should be stop the module clock, when change this bit<br><b>When using OTG_PHY, the UHCCDR should be = 0;</b>               | RW |
| 29        | CE_UHC   | Change enable for UHC. If CE_UHC is 1 , write on UHCCDR ill start a frequency changing sequence immediately. If CE_UHC is 0, writes on UHCCDR have no affect                                                                                                                                                             | RW |
| 28        | UHC_BUSY | The bit is read only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should wait until UHC_BUSY == 0, may begin another frequency change.                               | RW |
| 27        | UHC_STOP | When UHC_STOP is 1 and CE_UHC is 1, the UHC clock will stop.<br>When UHC_STOP is 0 and CE_UHC is 1, the UHC clock will continue.<br>Software should wait until UHC_BUSY == 0, may begin another frequency change.<br>UHC_STOP is prior to UHCCDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 26:8      | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                     | R  |
| 7:0       | UHCCDR   | Divider for UHC Frequency. Specified the UHC device clock division ratio<br>division ratio = UHCCDR + 1                                                                                                                                                                                                                  | RW |

### 18.1.2.12 SSI device clock divider Register

SSI device clock divider Register (SSICDR) is a 32-bit read/write register that specifies the divider of SSI device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on SSICDR.

| SSICDR |     |      |        |          |          |          |    |    |    |    |    |    |    |    |    | 0x10000074 |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
|--------|-----|------|--------|----------|----------|----------|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|
| Bit    | 31  | 30   | 29     | 28       | 27       | 26       | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10     | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|        | SCS | SPCS | CE_SSI | SSI_BUSY | SSI_STOP | Reserved |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    | SSICDR |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
| RST    | 0   | 0    | 0      | 0        | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                | RW |
|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31   | SCS      | SSI Clock Source Selection. Selects the SSI clock source between PLL output and pin EXCLK.<br>0: SSI clock source is EXCLK<br>1: SSI clock source is PLL output divided by SSICDR<br>If SCS == 0, the clock divider is gated, don't set CE_SSI<br>It is no glitch free mux. Software should be stop the module clock, when change this bit | R  |
| 30   | SPCS     | 0: select SCLK_A output<br>1: select MPLL output<br>It is no glitch free mux. Software should be stop the module clock, when change this bit                                                                                                                                                                                               | RW |
| 29   | CE_SSI   | Change enable for SSI. If CE_SSI is 1 , write on SSICDR will start a frequency changing sequence immediately. If CE_SSI is 0, writes on SSICDR have no affect                                                                                                                                                                              | RW |
| 28   | SSI_BUSY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until SSI_BUSY == 0, may begin another frequency change.                                             | RW |
| 27   | SSI_STOP | When SSI_STOP is 1 and CE_SSI is 1, the SSI clock will stop.<br>When SSI_STOP is 0 and CE_SSI is 1, the SSI clock will continue.<br>Software should wait until SSI_BUSY == 0, may begin another frequency change.<br><br>SSI_STOP is prior to SSICDR.<br>If system don't use this module, may set this bit to reduce power .               | RW |
| 26:8 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                       | R  |
| 7:0  | SSICDR   | Divider for SSI Frequency. Specified the SSI device clock division ratio, which varies from 1 to 256 (division ratio = SSICDR + 1).                                                                                                                                                                                                        | RW |

### 18.1.2.13 CIM MCLK clock divider Register

CIM mclk clock divider Register (CIMCDR) is a 32-bit read/write register that specifies the divider of CIM mclk clock (CIM\_MCLK). This register is initialized to 0x00000000 only by any reset. Only word access can be used on CIMCDR.

|               |        |        |          |          |          |    |    |    |    |    |    |    |    |    |    |                   |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|---------------|--------|--------|----------|----------|----------|----|----|----|----|----|----|----|----|----|----|-------------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| <b>CIMCDR</b> |        |        |          |          |          |    |    |    |    |    |    |    |    |    |    | <b>0x1000007C</b> |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>    | 31     | 30     | 29       | 28       | 27       | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|               | CIMPCS | CE_CIM | CIM_BUSY | CIM_STOP | Reserved |    |    |    |    |    |    |    |    |    |    |                   | CIMCDR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>    | 0      | 0      | 0        | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0      | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

| Bits | Name         | Description                                                                                                                                                                                                                                                                                                              | RW |
|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31   | CIMPCS       | 0: select SCLK_A clock output<br>1: select MPLL clock output<br>It is no glitch free mux. Software should be stop the module clock, when change this bit                                                                                                                                                                 | RW |
| 30   | CE_CIM       | Change enable for CIM. If CE_CIM is 1 , write on CIMCDR ill start a frequency changing sequence immediately. If CE_CIM is 0, writes on CIMCDR have no affect                                                                                                                                                             | RW |
| 29   | CIM_BU<br>SY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until CIM_BUSY == 0, may begin another frequency change.                           | RW |
| 28   | CIM_ST<br>OP | When CIM_STOP is 1 and CE_CIM is 1, the CIM clock will stop.<br>When CIM_STOP is 0 and CE_CIM is 1, the CIM clock will continue.<br>Software should wait until CIM_BUSY == 0, may begin another frequency change.<br>CIM_STOP is prior to CIMCDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 27:8 | Reserved     | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                     | R  |
| 7:0  | CIMCDR       | Divider for CIM MCLK Frequency. Specified the CIM MCLK clock (CIM_MCLK) division ratio, which varies from 1 to 256 (division ratio = CIMCDR + 1). the output clk is io_setup.cim_mclk_o                                                                                                                                  | RW |

### 18.1.2.14 PCM device clock divider Register

PCM device clock divider Register (PCMCDR) is a 32-bit read/write register that specifies the divider of PCM device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on PCMCDR.

| PCMCDR |      |        |        |          |          |          |    |    |    |    |    |    |    |    |    | 0x10000084 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|------|--------|--------|----------|----------|----------|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit    | 31   | 30     | 29     | 28       | 27       | 26       | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | PCMS | PCMPCS | CE_PCM | PCM_BUSY | PCM_STOP | Reserved |    |    |    |    |    |    |    |    |    | PCMCDR     |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST    | 0    | 0      | 0      | 0        | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                       | RW |
|-------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | PCMS     | PCM source clock Selection.<br>0: PCM source clock is pin EXCLK<br>1: PCM source clock is PLL divider output<br>It is no glitch free mux. Software should be stop the module clock, when change this bit                                                                                          | RW |
| 30:29 | PCMPCS   | 00: select SCLK_A output<br>01: select MPLL output<br>10: select EPLL output<br>11: select VPLL output<br>It is no glitch free mux. Software should be stop the module clock, when change this bit                                                                                                | RW |
| 28    | CE_PCM   | Change enable for PCM. If CE_PCM is 1 , write on PCMCDR ill start a frequency changing sequence immediately. If CE_PCM is 0, writes on PCMCDR have no affect                                                                                                                                      | RW |
| 27    | PCM_BUSY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until PCM_BUSY == 0, may go next                            | RW |
| 26    | PCM_STOP | When PCM_STOP is 1 and CE_PCM is 1, the PCM clock will stop.<br>When PCM_STOP is 0 and CE_PCM is 1, the PCM clock will continue.<br>Software should wait until PCM_BUSY == 0, may go next.<br>PCM_STOP is prior to PCMCDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 25:8  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                              | R  |
| 7:0   | PCMCDR   | Divider for PCM Frequency. Specified the PCM device clock division ratio, which varies from 1 to 256 (division ratio = PCMCDR + 1).                                                                                                                                                               | RW |

### 18.1.2.15 GPU clock divider Register

GPU clock divider Register (GPUCDR) is a 32-bit read/write register that specifies the divider of GPU clock . This register is initialized to 0x40000000 only by any reset. Only word access can be used on GPUCDR.

|               |      |    |    |    |        |          |          |          |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |        |   |   |   |   |   |   |
|---------------|------|----|----|----|--------|----------|----------|----------|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|--------|---|---|---|---|---|---|
| <b>GPUCDR</b> |      |    |    |    |        |          |          |          |    |    |    |    |    |    |    | <b>0x10000088</b> |    |    |    |    |    |    |   |   |   |   |        |   |   |   |   |   |   |
| <b>Bit</b>    | 31   | 30 | 29 | 28 | 27     | 26       | 25       | 24       | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5      | 4 | 3 | 2 | 1 | 0 |   |
|               | GPCS |    |    |    | CE_GPU | GPU_BUSY | GPU_STOP | Reserved |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   | GPUCDR |   |   |   |   |   |   |
| <b>RST</b>    | 0    | 1  | 0  | 0  | 0      | 0        | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                              | RW |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | GPCS     | 00: stop clock mux output<br>01: select SCLK_A output<br>10: select MPLL output<br>11: select EPLL output<br>The clock mux is free of glitches if clock selection is changed                                                                                                                                             | RW |
| 29    | CE_GPU   | Change enable for GPU. If CE_GPU is 1 , write on GPUCDR ill start a frequency changing sequence immediately. If CE_GPU is 0, writes on GPUCDR have no affect                                                                                                                                                             | RW |
| 28    | GPU_BUSY | The bit is read only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until GPU_BUSY == 0, may begin another frequency change.                            | RW |
| 27    | GPU_STOP | When GPU_STOP is 1 and CE_GPU is 1, the GPU clock will stop.<br>When GPU_STOP is 0 and CE_GPU is 1, the GPU clock will continue.<br>Software should wait until GPU_BUSY == 0, may begin another frequency change.<br>GPU_STOP is prior to GPUCDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 26:4  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                     | R  |
| 3:0   | GPUCDR   | Divider for GPU Frequency. Specified the GPU clock division ratio, which varies from 1 to 16 (division ratio = GPUCDR + 1).                                                                                                                                                                                              | RW |

### 18.1.2.16 HDMI clock divider Register

HDMI clock divider Register (HDMICDR) is a 32-bit read/write register that specifies the divider of HDMI clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on HDMICDR.

| HDMICDR |      |    |    |    |         |           |           |          |    |    |    |    |    |    |    | 0x100008C |    |         |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|---------|------|----|----|----|---------|-----------|-----------|----------|----|----|----|----|----|----|----|-----------|----|---------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit     | 31   | 30 | 29 | 28 | 27      | 26        | 25        | 24       | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16        | 15 | 14      | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|         | HPCS |    |    |    | CE_HDMI | HDMI_BUSY | HDMI_STOP | Reserved |    |    |    |    |    |    |    |           |    | HDMICDR |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST     | 0    | 0  | 0  | 0  | 0       | 0         | 0         | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0         | 0  | 0       | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name      | Description                                                                                                                                                                                                                                                                                                                   | RW |
|-------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | HPCS      | 00: select SCLK_A output<br>01: select MPLL output<br>10: select VPLL output<br>It is no glitch free mux. Software should be stop the module clock, when change this bit                                                                                                                                                      | RW |
| 29    | CE_HDMI   | Change enable for HDMI. If CE_HDMI is 1 , write on HDMICDR ill start a frequency changing sequence immediatly. If CE_HDMI is 0, writes on HDMICDR have no affect                                                                                                                                                              | RW |
| 28    | HDMI_BUSY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until HDMI_BUSY == 0, may begin another frequency change.                               | RW |
| 26    | HDMI_STOP | When HDMI_STOP is 1 and CE_HDMI is 1, the HDMI clock will stop.<br>When HDMI_STOP is 0 and CE_HDMI is 1, the HDMI clock will continue.<br>Software should wait until HDMI_BUSY == 0, may begin another frequency change.HDMI_STOP is prior to HDMICDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 25:8  | Reserved  | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                          | R  |
| 7:0   | HDMICDR   | Divider for HDMI Frequency. Specified the HDMI clock division ratio, which varies from 1 to 256 (division ratio = HDMICDR + 1).                                                                                                                                                                                               | RW |

**Import Note:** The clock must be between 18M and 27M, it occurs fatal error if exceeding the range.

### 18.1.2.17 BCH clock divider Register

BCH clock divider Register (BCHCDR) is a 32-bit read/write register that specifies the divider of BCH clock . This register is initialized to 0x40000000 only by any reset. Only word access can be used on BCHCDR.

|               |      |    |    |    |    |        |    |    |    |    |          |    |    |    |    |                   |    |    |    |    |          |    |   |   |   |   |   |   |   |   |   |   |   |   |   |  |        |  |  |  |
|---------------|------|----|----|----|----|--------|----|----|----|----|----------|----|----|----|----|-------------------|----|----|----|----|----------|----|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--------|--|--|--|
| <b>GPUCDR</b> |      |    |    |    |    |        |    |    |    |    |          |    |    |    |    | <b>0x100000AC</b> |    |    |    |    |          |    |   |   |   |   |   |   |   |   |   |   |   |   |   |  |        |  |  |  |
| <b>Bit</b>    | 31   | 30 | 29 | 28 | 27 | 26     | 25 | 24 | 23 | 22 | 21       | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |   |  |        |  |  |  |
|               | BPCS |    |    |    |    | CE_BCH |    |    |    |    | BCH_BUSY |    |    |    |    | BCH_STOP          |    |    |    |    | Reserved |    |   |   |   |   |   |   |   |   |   |   |   |   |   |  | BCHCDR |  |  |  |
| <b>RST</b>    | 0    | 1  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0                 | 0  | 0  | 0  | 0  | 0        | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |        |  |  |  |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                          | RW |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | BPCS     | 00: stop clock mux output<br>01: select SCLK_A output<br>10: select MPLL output<br>11: select EPLL output<br>The clock mux is free of glitches if clock selection is changed<br>When switching clock source, it should be ensured that the clock switched from and the clock switched to are running.                | RW |
| 29    | CE_BCH   | Change enable for BCH. If CE_BCH is 1 , write on BCHCDR ill start a frequency changing sequence immediately. If CE_BCH is 0, writes on BCHCDR have no affect                                                                                                                                                         | RW |
| 28    | BCH_BUSY | The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going<br>Software should be wait until BCH_BUSY == 0, may begin another frequency change.                       | RW |
| 27    | BCH_STOP | When BCH_STOP is 1 and CE_BCH is 1, the BCH clock will stop.<br>When BCH_STOP is 0 and CE_BCH is 1, the BCH clock will continue.<br>Software should wait until BCH_BUSY == 0, may begin another frequency change.BCH_STOP is prior to BCHCDR.<br>If system don't use this module, may set this bit to reduce power . | RW |
| 26:4  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                 | R  |
| 3:0   | BCHCDR   | Divider for BCH Frequency. Specified the BCH clock division ratio, which varies from 1 to 16 (division ratio = BCHCDR + 1).                                                                                                                                                                                          | RW |

### 18.1.2.18 CPM interrupt Register

|                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>CPM_INTR</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x100000B0</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>      | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |



|          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |           |           |   |
|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----------|-----------|---|
| Reserved |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | VBUS_INTR | ADEV_INTR |   |
| RST      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0         | 0         | 0 |

| Bits | Name          | Description                          | RW |
|------|---------------|--------------------------------------|----|
| 31:3 | Reserved      | Writing has no effect, read as zero. | R  |
| 1    | VBUS_IN<br>TR | B device insert interrupt.           | R  |
| 0    | ADEV_IN<br>TR | A device insert interrupt.           | R  |

### 18.1.2.19 CPM interrupt enable Register

|           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |    |   |   |   |   |   |            |            |   |   |   |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|---|---|---|---|---|------------|------------|---|---|---|
| CPM_INTRE |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10000B4 |    |    |    |    |    |    |   |   |   |   |   |            |            |   |   |   |
| Bit       | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16        | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4          | 3          | 2 | 1 | 0 |
| Reserved  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |    |   |   |   |   |   | VBUS_INTRE | ADEV_INTRE |   |   |   |
| RST       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0          | 0          | 0 | 0 |   |

| Bits | Name           | Description                          | RW |
|------|----------------|--------------------------------------|----|
| 31:3 | Reserved       | Writing has no effect, read as zero. | R  |
| 1    | VBUS_IN<br>TRE | B device insert interrupt enable.    | RW |
| 0    | ADEV_IN<br>TRE | A device insert interrupt enable.    | RW |

### 18.1.2.20 CPM Scratch Pad Protected Register

The Scratch Pad Protected Register is reset to 0x0000a5a5. When CPSPPR value equals to 0x00005a5a, software can write the CPSPPR. or else can't.

|          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| CPSPPR   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x1000038 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit      | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16        | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CPSPPR    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0         | 1  | 0  | 1  | 0  | 0  | 1  | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

| Bits  | Name     | Description                                                   | RW |
|-------|----------|---------------------------------------------------------------|----|
| 31:16 | Reserved | Writing has no effect, read as zero.                          | R  |
| 15:0  | CPSPPR   | The value is only = 0x00005a5a, software can write the CPSPR. | RW |

### 18.1.2.21 CPM Scratch Pad Register

The Scratch Pad Register is a 32-bit read/write register that allows software to preserve some critical data . It is not initialized by power on and WDT reset.

| CPPSR |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10000034 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31    | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | CPPSR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST   | ?     | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?          | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

### 18.1.2.22 USB Parameter Control Register

The USBPCR is a 32-bit read/write register that allows software to control OTG PHY some functions. It is initialized to 0x409919b8.

| USBPCR |          |          |               |           |            |          |              |     |       |            |             |         |          |            |            | 0x1000003C |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|----------|----------|---------------|-----------|------------|----------|--------------|-----|-------|------------|-------------|---------|----------|------------|------------|------------|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| Bit    | 31       | 30       | 29            | 28        | 27         | 26       | 25           | 24  | 23    | 22         | 21          | 20      | 19       | 18         | 17         | 16         | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
|        | USB_MODE | AVLD_REG | IDPULLUP_MASK | INCR_MASK | TXRISETUNE | COMONONN | VBUSVLDXTSEL | POR | SIDDQ | OTGDISABLE | COMPDISTUNE | OTGTUNE | SQRXTUNE | TXFSLSTUNE | TXPREMTUNE | TXHSXVTUNE | TXVREFTUNE |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| RST    | 0        | 1        | 0             | 0         | 0          | 0        | 0            | 1   | 0     | 0          | 1           | 1       | 0        | 0          | 1          | 0          | 0          | 0  | 1  | 1  | 0  | 0  | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |  |

| Bits  | Name          | Description                                                                                                                                                                             | RW |
|-------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | USB_MODE      | 0: work as USB device<br>1: work as OTG                                                                                                                                                 | RW |
| 30    | AVLD_REG      | This bit is used to set “avalid”(VBUS above A-device session threshold) signal.                                                                                                         | RW |
| 29:28 | IDPULLUP_MASK | These 2 bits control “idpullup” signal in otg mode.<br>2'b1x: “idpullup” always active<br>2'b01: “idpullup” always active when usb suspend<br>2'b00: use “idpullup” from otg controller | RW |
| 27    | INCR_MASK     | This bit controls whether the ahb interface enhancement for “incr transfer” takes effect. Set this bit to 0 will active the enhancement.                                                | RW |
| 26    | TXRISETUNE    | This bit adjusts the rise/fall times of the high-speed waveform                                                                                                                         | RW |

|       |               |                                                                                                                                                                                 |         |
|-------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
|       |               | 1: -8%<br>0: default                                                                                                                                                            |         |
| 25    | COMMONONN     | This bit is the OTG PHY common block power down control signal.<br>0: The common blocks remain powered in suspend mode<br>1: The common blocks are powered down in suspend mode | RW      |
| 24    | VBUSVLDEXT    | This bit controls OTG PHY VBUSVLDEXT signal.                                                                                                                                    | RW      |
| 23    | VBUSVLDEXTSEL | This bit controls OTG PHY VBUSVLDEXTSEL signal.                                                                                                                                 | RW      |
| 22    | POR           | This bit controls OTG PHY power on reset.                                                                                                                                       | RW      |
| 21    | SIDDQ         | This bit is the OTG PHY analog blocks power down signal.                                                                                                                        | RW      |
| 20    | OTG_DISABLE   | This bit is the power control for otg block in OTG PHY.                                                                                                                         | RW      |
| 19:17 | COMPDISTUNE   | These bits control disconnect threshold adjustment.                                                                                                                             | RW      |
|       |               | 3'b111                                                                                                                                                                          | +4.5%   |
|       |               | 3'b110                                                                                                                                                                          | +3%     |
|       |               | 3'b101                                                                                                                                                                          | +1.5%   |
|       |               | 3'b100                                                                                                                                                                          | Default |
|       |               | 3'b011                                                                                                                                                                          | -1.5%   |
|       |               | 3'b010                                                                                                                                                                          | -3%     |
|       |               | 3'b001                                                                                                                                                                          | -4.5%   |
|       |               | 3'b000                                                                                                                                                                          | -6%     |
| 16:14 | OTGTUNE       | These bits control VBUS valid threshold adjustment.                                                                                                                             | RW      |
|       |               | 3'b111                                                                                                                                                                          | +4.5%   |
|       |               | 3'b110                                                                                                                                                                          | +3%     |
|       |               | 3'b101                                                                                                                                                                          | +1.5%   |
|       |               | 3'b100                                                                                                                                                                          | Default |
|       |               | 3'b011                                                                                                                                                                          | -1.5%   |
|       |               | 3'b010                                                                                                                                                                          | -3%     |
|       |               | 3'b001                                                                                                                                                                          | -4.5%   |
|       |               | 3'b000                                                                                                                                                                          | -6%     |
| 13:11 | SQRXTUNE      | These bits control squelch threshold adjustment.                                                                                                                                | RW      |
|       |               | 3'b111                                                                                                                                                                          | -20%    |
|       |               | 3'b110                                                                                                                                                                          | -15%    |
|       |               | 3'b101                                                                                                                                                                          | -10%    |
|       |               | 3'b100                                                                                                                                                                          | -5%     |
|       |               | 3'b011                                                                                                                                                                          | default |
|       |               | 3'b010                                                                                                                                                                          | +5%     |
|       |               | 3'b001                                                                                                                                                                          | +10%    |
|       |               | 3'b000                                                                                                                                                                          | +15%    |
| 10:7  | TXFSLSTUNE    | These bits control FS/LS source impedance adjustment.                                                                                                                           | RW      |
|       |               | 4'b1111                                                                                                                                                                         | -5%     |
|       |               | 4'b0111                                                                                                                                                                         | -2.5%   |
|       |               | 4'b0011                                                                                                                                                                         | Default |

|     |               |                                                                                               |          |    |
|-----|---------------|-----------------------------------------------------------------------------------------------|----------|----|
|     |               | 4'b0001                                                                                       | +2.5%    |    |
|     |               | 4'b0000                                                                                       | +5%      |    |
| 6   | TXPREEMPHTUNE | This bit controls HS transmitter Pre-emphasis enable.<br>1: enable<br>0: disable              |          | RW |
| 5:4 | TXHSXVTUNE    | These bits adjust the voltage at which dp and dm signals cross while transmitting in HS mode. |          | RW |
|     |               | 2'b11                                                                                         | Default  |    |
|     |               | 2'b10                                                                                         | +15mv    |    |
|     |               | 2'b01                                                                                         | -15mv    |    |
|     |               | 2'b00                                                                                         | reserved |    |
| 3:0 | TXVREFTUNE    | These bits control HS DC voltage level adjustment.                                            |          | RW |
|     |               | 4'b1111                                                                                       | +12.5%   |    |
|     |               | 4'b1110                                                                                       | +11.25%  |    |
|     |               | 4'b1101                                                                                       | +10%     |    |
|     |               | 4'b1100                                                                                       | +8.75%   |    |
|     |               | 4'b1011                                                                                       | +7.5%    |    |
|     |               | 4'b1010                                                                                       | +6.255   |    |
|     |               | 4'b1001                                                                                       | +5%      |    |
|     |               | 4'b1000                                                                                       | +3.75%   |    |
|     |               | 4'b0111                                                                                       | +2.5%    |    |
|     |               | 4'b0110                                                                                       | +1.25%   |    |
|     |               | 4'b0101                                                                                       | Default  |    |
|     |               | 4'b0100                                                                                       | -1.25%   |    |
|     |               | 4'b0011                                                                                       | -2.5%    |    |
|     |               | 4'b0010                                                                                       | -3.75%   |    |
|     |               | 4'b0001                                                                                       | -5%      |    |
|     |               | 4'b0000                                                                                       | -6.25%   |    |

### 18.1.2.23 USB Reset Detect Timer Register

It is initialized to 0x2000096

|            |               |    |    |    |    |         |             |          |           |        |    |    |    |    |    |    |                   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|------------|---------------|----|----|----|----|---------|-------------|----------|-----------|--------|----|----|----|----|----|----|-------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
|            | <b>USBRDT</b> |    |    |    |    |         |             |          |           |        |    |    |    |    |    |    | <b>0x10000040</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31            | 30 | 29 | 28 | 27 | 26      | 25          | 24       | 23        | 22     | 21 | 20 | 19 | 18 | 17 | 16 | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|            | Reserved      |    |    |    |    | HB_MASK | VBFIL_LD_EN | IDDIG_EN | IDDIG_REG | USBRDT |    |    |    |    |    |    |                   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b> | ?             | ?  | ?  | ?  | ?  | 0       | 1           | 0        | 0         | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |

| Bits  | Name        | Description                                                   | RW |
|-------|-------------|---------------------------------------------------------------|----|
| 31:27 | Reserved    | Writing has no effect, read as zero.                          | R  |
| 26    | HB_MASK     | Halfword/Byte transfer support mask.<br>0: enable<br>1: mask  | RW |
| 25    | VBFIL_LD_EN | VBUS filter data load enable.                                 | RW |
| 24    | IDDIG_EN    | This bit indicates using IDDIG_REG to control "iddig" signal. | RW |
| 23    | IDDIG_REG   | This bit controls "iddig" when IDDIG_REG_EN = 1'b1.           | RW |
| 22:0  | USBRDT      | These bits control USB reset detect time.                     | RW |

### 18.1.2.24 USB VBUS Jitter Filter Register

| USBVBFIL |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10000044 |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit      | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|          | IDDIGFIL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | USBVBFIL |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST      | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1          | 0        | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                   | RW |
|-------|----------|-----------------------------------------------|----|
| 31:16 | IDDIGFIL | These bits controls iddig jitter filter time. | RW |
| 15:0  | USBVBFIL | These bits controls VBUS jitter filter time.  | RW |

### 18.1.2.25 USB Parameter Control Register1

The USBPCR1 is a 32-bit read/write register that allows software to control OTG PHY some functions. It is initialized to 0x0cc23370.

| USBPCR1 |          |    |    |         |           |           |       |       |           |           |          |          |              |           |             | 0x10000048     |             |             |             |    |    |    |   |   |   |   |   |   |   |   |   |   |
|---------|----------|----|----|---------|-----------|-----------|-------|-------|-----------|-----------|----------|----------|--------------|-----------|-------------|----------------|-------------|-------------|-------------|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit     | 31       | 30 | 29 | 28      | 27        | 26        | 25    | 24    | 23        | 22        | 21       | 20       | 19           | 18        | 17          | 16             | 15          | 14          | 13          | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|         | reserved |    |    | USB_SEL | REFCLKSEL | REFCLKDIV | DMPD1 | DPPD1 | PORT0_RST | PORT1_RST | WORD_IF0 | WORD_IF1 | COMPDISTUNE1 | SQRXTUNE1 | TXFSLSTUNE1 | TXPREEMPHTUNE1 | TXHSXVTUNE1 | TXVREFTUNE1 | TXRISETUNE1 |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST     | ?        | ?  | ?  | 0       | 1         | 1         | 0     | 0     | 1         | 1         | 0        | 0        | 0            | 0         | 1           | 0              | 0           | 0           | 1           | 1  | 0  | 0  | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bits | Name        | Description                      | RW |
|------|-------------|----------------------------------|----|
| 28   | USB_SE<br>L | 1: synopsys otg<br>0: mentor otg |    |

|        |               |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
|--------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------|--------|-----|--------|-------|--------|---------|--------|-------|--------|-----|--------|-------|--------|-----|--|
| 27:26  | REFCLK SEL    | This bus selects the reference clock source<br>11, 10: The PLL uses CLKCORE as reference<br>01: The XO block uses an external, 2.5v clock supplied on the XO pin<br>00: The XO block uses the clock from a crystal.                                                                                                                                                                                                                   |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 25:24  | REFCLK DIV    | This bus selects the reference clock frequency<br>11: 19.2MHz<br>10: 48MHz<br>01: 24MHz<br>00: 12MHz                                                                                                                                                                                                                                                                                                                                  |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 23     | DMPD1         | 1: The pull-down resistance on D- of port1 is enabled<br>0: The pull-down resistance on D- of port1 is disabled.                                                                                                                                                                                                                                                                                                                      |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 22     | DPPD1         | 1: The pull-down resistance on D+ of port1 is enabled<br>0: The pull-down resistance on D+ of port1 is disabled.                                                                                                                                                                                                                                                                                                                      |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 21     | PORT0_ RST    | Port0 reset:<br>1: The transmit and receive finite state machines are reset, and the linstate logic combinatorially reflects the state of the single-ended receivers<br>0: The transmit and receive finite machines are operational, and the linstate inputs becomes sequential after 11 PHYCLOCK cycles.                                                                                                                             |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 20     | PORT1_ RST    | Port1 reset:<br>1: The transmit and receive finite state machines are reset, and the linstate logic combinatorially reflects the state of the single-ended receivers<br>0: The transmit and receive finite machines are operational, and the linstate inputs becomes sequential after 11 PHYCLOCK cycles.                                                                                                                             |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 19     | WORD_I F0     | This bit selects utmi data bus width of port0<br>1: 16bit/30M<br>0: 8bit/60M                                                                                                                                                                                                                                                                                                                                                          |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 18     | WORD_I F1     | This bit selects utmi data bus width of port1.<br>1: 16bit/30M<br>0: 8bit/60M                                                                                                                                                                                                                                                                                                                                                         |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 17:15  | COMPDI STUNE1 | These bits control disconnect threshold adjustment.<br><table border="1" data-bbox="448 1565 1315 1912"> <tr><td>3'b111</td><td>+4.5%</td></tr> <tr><td>3'b110</td><td>+3%</td></tr> <tr><td>3'b101</td><td>+1.5%</td></tr> <tr><td>3'b100</td><td>Default</td></tr> <tr><td>3'b011</td><td>-1.5%</td></tr> <tr><td>3'b010</td><td>-3%</td></tr> <tr><td>3'b001</td><td>-4.5%</td></tr> <tr><td>3'b000</td><td>-6%</td></tr> </table> | 3'b111 | +4.5% | 3'b110 | +3% | 3'b101 | +1.5% | 3'b100 | Default | 3'b011 | -1.5% | 3'b010 | -3% | 3'b001 | -4.5% | 3'b000 | -6% |  |
| 3'b111 | +4.5%         |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 3'b110 | +3%           |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 3'b101 | +1.5%         |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 3'b100 | Default       |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 3'b011 | -1.5%         |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 3'b010 | -3%           |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 3'b001 | -4.5%         |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 3'b000 | -6%           |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 14:12  | SQRXTU NE1    | These bits control squelch threshold adjustment.<br><table border="1" data-bbox="448 1955 1315 1995"> <tr><td>3'b111</td><td>-20%</td></tr> </table>                                                                                                                                                                                                                                                                                  | 3'b111 | -20%  |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |
| 3'b111 | -20%          |                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |       |        |     |        |       |        |         |        |       |        |     |        |       |        |     |  |

|      |                |                                                                                               |          |    |
|------|----------------|-----------------------------------------------------------------------------------------------|----------|----|
|      |                | 3'b110                                                                                        | -15%     |    |
|      |                | 3'b101                                                                                        | -10%     |    |
|      |                | 3'b100                                                                                        | -5%      |    |
|      |                | 3'b011                                                                                        | default  |    |
|      |                | 3'b010                                                                                        | +5%      |    |
|      |                | 3'b001                                                                                        | +10%     |    |
|      |                | 3'b000                                                                                        | +15%     |    |
| 11:8 | TXFSLSTUNE1    | These bits control FS/LS source impedance adjustment.                                         |          | RW |
|      |                | 4'b1111                                                                                       | -5%      |    |
|      |                | 4'b0111                                                                                       | -2.5%    |    |
|      |                | 4'b0011                                                                                       | Default  |    |
|      |                | 4'b0001                                                                                       | +2.5%    |    |
|      |                | 4'b0000                                                                                       | +5%      |    |
| 7    | TXPREEMPHTUNE1 | This bit controls HS transmitter Pre-emphasis enable.<br>1: enable<br>0: disable              |          | RW |
| 6:5  | TXHSXVTUNE1    | These bits adjust the voltage at which dp and dm signals cross while transmitting in HS mode. |          | RW |
|      |                | 2'b11                                                                                         | Default  |    |
|      |                | 2'b10                                                                                         | +15mv    |    |
|      |                | 2'b01                                                                                         | -15mv    |    |
|      |                | 2'b00                                                                                         | reserved |    |
| 4:1  | TXVREFTUNE1    | These bits control HS DC voltage level adjustment.                                            |          | RW |
|      |                | 4'b1111                                                                                       | +12.5%   |    |
|      |                | 4'b1110                                                                                       | +11.25%  |    |
|      |                | 4'b1101                                                                                       | +10%     |    |
|      |                | 4'b1100                                                                                       | +8.75%   |    |
|      |                | 4'b1011                                                                                       | +7.5%    |    |
|      |                | 4'b1010                                                                                       | +6.25%   |    |
|      |                | 4'b1001                                                                                       | +5%      |    |
|      |                | 4'b1000                                                                                       | +3.75%   |    |
|      |                | 4'b0111                                                                                       | +2.5%    |    |
|      |                | 4'b0110                                                                                       | +1.25%   |    |
|      |                | 4'b0101                                                                                       | Default  |    |
|      |                | 4'b0100                                                                                       | -1.25%   |    |
|      |                | 4'b0011                                                                                       | -2.5%    |    |
|      |                | 4'b0010                                                                                       | -3.75%   |    |
|      |                | 4'b0001                                                                                       | -5%      |    |
|      |                | 4'b0000                                                                                       | -6.25%   |    |
| 0    | TXRISETUNE1    | This bit adjusts the rise/fall times of the high-speed waveform<br>1: -8%<br>0: default       |          | RW |

**Note:**

\*1: port1 is uhc, port 0 is otg

**18.1.2.26 PLL Control Register**

The PLL Control Register (CPPCR) is a 32-bit read/write register, which controls PLL some controls bits and stabilize time. It is initialized to 0x80000020 only by any reset. Only word access can be used on CPPCR. It is used to control all PLL.

| CPPCR |       |        |          |    |    |    |    |    |    |    |       |    |    |    |    | 0x1000000C |    |    |       |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|-------|--------|----------|----|----|----|----|----|----|----|-------|----|----|----|----|------------|----|----|-------|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31    | 30     | 29       | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21    | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13    | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | ENSAT | FASTEN | Reserved |    |    |    |    |    |    |    | BWADJ |    |    |    |    |            |    |    | PLLST |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST   | 1     | 0      | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0     | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                      | RW |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | ENSAT    | ENSAT should be tied 1 in normal operation. Setting ENSAT high ensures that all critical analog circuitry are biased at their optimal values across all PLL operating ranges. ENSAT only affects NF values typically less than or equal to 32 (depending on the exact design version). Only under very special conditions should ENSAT be set low.<br><b>Don't change the value</b>                              | RW |
| 30    | FASTEN   | FASTEN controls a fast locking behavior mode for the PLL, but does not affect normal locked operation. Setting FASTEN high will reduce the lock time to less than:<br>lock time = (NR * 500) reference cycles<br>FASTEN should be tied 0 by default, but should be controlled by a register for testing and specific application modes.<br><b>Don't change the value</b>                                         | RW |
| 29:28 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                             | R  |
| 27:26 | Reserved |                                                                                                                                                                                                                                                                                                                                                                                                                  | R  |
| 19:8  | BWADJ    | A 12-bit bus that selects the values 1-4096 for the bandwidth divider (NB)<br>NB = BWADJ[11:0] + 1<br>The loop bandwidth (BW) of the PLL can be adjusted using BWADJ[11:0].<br>The bandwidth is given by:<br>BW = nom_BW*sqrt(NF / 2 / NB)<br>where nom_BW is approximately given by:<br>nom_BW = Fref / (NR*20)<br>and Fref is the reference clock frequency. The damping factor (D) is approximately given by: | RW |



|     |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |    |
|-----|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|     |       | $D = \text{nom\_D} \cdot \sqrt{NF / 2 / NB}$ <p>where nom_D is approximately 1. Because the damping factor changes with bandwidth settings, the bandwidth is practically limited to:</p> $\text{nom\_BW} / \sqrt{2} < BW < \text{nom\_BW} \cdot \sqrt{2}$ <p>in order to limit the damping factor range to 0.7 - 1.4. The -3dB bandwidth (Fbw_3dB) is approximately given by:</p> $Fbw\_3dB = 2.4 * \text{nom\_BW} * (NF / 2 / NB)$ <p>The recommended setting for NB is NF / 2, which will yield the nominal bandwidth. Note that nom_BW and nom_D are chosen to result in optimal PLL loop dynamics.</p> <p><b>The value can only be used to debug</b></p> |    |
| 7:0 | PLLST | PLL Stabilize Time. Specifies the PLL stabilize time by unit of RTCCLK (approximate 32kHz) cycles. It is used when change PLL multiplier or change PLL from off to on. It is initialized to H'20. It is used to control all PLL.                                                                                                                                                                                                                                                                                                                                                                                                                             | RW |

### 18.1.2.27 APLL Control Register

The APLL Control Register (CPAPCR) is a 32-bit read/write register, which controls PLL multiplier, on/off state and stabilize time. It is initialized to 0x00000000 only by any reset. Only word access can be used on CPAPCR.

| CPAPCR |       |    |    |    |    |    |    |    |    |    |    |    |       |    |    | 0x10000010 |    |    |        |    |    |    |         |         |       |       |         |         |       |       |       |   |   |
|--------|-------|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|------------|----|----|--------|----|----|----|---------|---------|-------|-------|---------|---------|-------|-------|-------|---|---|
| Bit    | 31    | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19    | 18 | 17 | 16         | 15 | 14 | 13     | 12 | 11 | 10 | 9       | 8       | 7     | 6     | 5       | 4       | 3     | 2     | 1     | 0 |   |
|        | APLLM |    |    |    |    |    |    |    |    |    |    |    | APLLN |    |    |            |    |    | APLLOD |    |    |    | ARFSLIP | AFBSLIP | ALOCK | ATEST | APLL_ON | AF_MODE | ABW_M | PLLBP | APLEN |   |   |
| RST    | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0          | 0  | 0  | 0      | 0  | 0  | 0  | 0       | 0       | 0     | 0     | 0       | 0       | 0     | 0     | 0     | 0 | 0 |

| Bits  | Name    | Description                                                                                                                                                                                      | RW |
|-------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:19 | APLLM   | APLL feedback 13-bit multiplication factor<br>$NF = \text{APLLM}[12:0] + 1$                                                                                                                      | RW |
| 18:13 | APLLN   | APLL input reference 6-bit divider.<br>$NR = \text{APLLN}[5:0] + 1$                                                                                                                              | RW |
| 12:9  | APLLOD  | APLL post VCO divider<br>$OD = \text{APLLOD}[3:0] + 1$ , this bit should always be odd number because vco doesn't support odd frequency divide. but OD can be 1.                                 | RW |
| 8     | ARFSLIP | ARFSLIP goes active for one or more divided feedback VCO cycles when the phase detector misses a divided reference cycle, i.e. when the VCO is running too fast. <b>It is only used to debug</b> | R  |
| 7     | AFBSLIP | FBSLIP goes active for one or more divided reference cycles when the                                                                                                                             | R  |

|   |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |    |
|---|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|   |         | phase detector misses a divided feedback VCO cycle, i.e. when the VCO is running too slow.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |    |
| 6 | ALOCK   | 0: the PLL output is not stable<br>1: the PLL output is stable<br>Software should clear this bit to 0, when this bit equal to 1, it indicates that PLL hadn't stable previously , <b>it is only used to debug.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | R  |
| 5 | AATEST  | Test PLL. It is only used to debug<br><b>Don't change the bit</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW |
| 4 | APLL_ON | PLL ON Flag.<br>0: PLL is off or not stable<br>1: PLL is on and stable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | R  |
| 3 | AF_MODE | 0: usually frequency change mode<br>1: Synchronous frequency change mode<br>Such a counter change can be performed only if the initial counter values and the changed counter values lead to valid operating frequencies.<br>Otherwise, the PLL could lose lock. If the change in the VCO frequency is less than $F_{bw\_3dB} * N_R / F_{ref}$ (where $F_{bw\_3dB}$ is the 3db bandwidth), the PLL will linearly settle (smoothly transition) to the new frequency. This constraint usually means that the frequency should be changed by less than 20%. Changes in excess of 20% can result in large frequency transients, which may force the PLL to lose lock. This state may necessitate putting the PLL through the reset sequence to re-achieve lock. Each linear settling change will require on the order of $1/F_{bw}$ (where $F_{bw}$ is the loop bandwidth), to mostly settle out. This approach is not suitable for simultaneous changes on PLLM and PLLN, as the approach does not guarantee that the PLLM and PLLN values will be observed on the same phase detection cycle | RW |
| 2 | ABW_M   | 0: BWADJ is controlled by hardware<br>1: BWADJ is controlled by software<br><b>Don't change the bit. it is only used to debug.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW |
| 1 | PLLBP   | PLL Bypass.<br>0: PLL output<br>1: EXCLK output<br><b>Don't change the bit. it is only used to debug.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW |
| 0 | APLLEN  | APLL Enable. When APLLEN is set to 1, APLL starts to lock phase.<br>After APLL stabilizes, APLL_ON bit is set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW |

### 18.1.2.28 MPLL Control Register

The MPLL Control Register (CPMPCR) is a 32-bit read/write register, which controls PLL multiplier, on/off state and stabilize time. It is initialized to 0x00000000 only by any reset. Only word access can be used on CPMPCR.

| CPMPCR |        |    |    |    |    |    |    |    |    |    |    |    |    | 0x10000014 |    |    |    |    |    |          |    |    |   |          |          |       |       |           |             |        |        |        |
|--------|--------|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----------|----|----|---|----------|----------|-------|-------|-----------|-------------|--------|--------|--------|
| Bit    | 31     | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18         | 17 | 16 | 15 | 14 | 13 | 12       | 11 | 10 | 9 | 8        | 7        | 6     | 5     | 4         | 3           | 2      | 1      | 0      |
|        | MPLL M |    |    |    |    |    |    |    |    |    |    |    |    | MPLL N     |    |    |    |    |    | MPLL O D |    |    |   | MRFS LIP | MFBS LIP | MLOCK | MTEST | MPLL_ O N | MF_ M O D E | MBW_ M | MPLLBP | MPLLEN |
| RST    | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0 | 0        | 0        | 0     | 0     | 0         | 0           | 0      | 0      | 0      |

| Bits  | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW |
|-------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:19 | MPLL M      | MPLL feedback 13-bit multiplication factor<br>NF = MPLLM[12:0] + 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW |
| 18:13 | MPLL N      | MPLL input reference 6-bit divider.<br>NR = MPLLN[5:0] + 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW |
| 12:9  | MPLL O D    | MPLL post VCO divider<br>OD = MPLLOD[3:0] + 1 ,this bit should always be odd number because vco doesn't support odd frequency divide. but OD can be 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW |
| 8     | MRFS LIP    | MRFS LIP goes active for one or more divided feedback VCO cycles when the phase detector misses a divided reference cycle, i.e. when the VCO is running too fast. <b>It is only used to debug</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | R  |
| 7     | MFBS LIP    | MFBS LIP goes active for one or more divided reference cycles when the phase detector misses a divided feedback VCO cycle, i.e. when the VCO is running too slow.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | R  |
| 6     | MLOCK       | 0: the PLL output is not stable<br>1: the PLL output is stable<br>Software should clear this bit to 0, when this bit equal to 1, it indicates that PLL hadn't stable previously , <b>it is only used to debug.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                 | R  |
| 5     | MTEST       | Test PLL. It is only used to debug<br><b>Don't change the bit</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW |
| 4     | MPLL_ O N   | PLL ON Flag.<br>0: PLL is off or not stable<br>1: PLL is on and stable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | R  |
| 3     | MF_ M O D E | 0: usually frequency change mode<br>1: Synchronous frequency change mode<br>Such a counter change can be performed only if the initial counter values and the changed counter values lead to valid operating frequencies.<br>Otherwise, the PLL could lose lock. If the change in the VCO frequency is less than $Fbw\_3dB * NR / Fref$ (where Fbw_3db is the 3db bandwidth), the PLL will linearly settle (smoothly transition) to the new frequency.<br>This constraint usually means that the frequency should be changed by less than 20%. Changes in excess of 20% can result in large frequency transients, which may force the PLL to lose lock. This state | RW |

|   |        |                                                                                                                                                                                                                                                                                                                                                                                                     |    |
|---|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|   |        | may necessitate putting the PLL through the reset sequence to re-achieve lock. Each linear settling change will require on the order of 1/Fbw (where Fbw is the loop bandwidth), to mostly settle out. This approach is not suitable for simultaneous changes on PLLM and PLLN, as the approach does not guarantee that the PLLM and PLLN values will be observed on the same phase detection cycle |    |
| 2 | MBW_M  | 0: BWADJ is controlled by hardware<br>1: BWADJ is controlled by software<br><b>Don't change the bit.</b>                                                                                                                                                                                                                                                                                            | RW |
| 1 | MPLLBP | PLL Bypass.<br>0: PLL output<br>1: EXCLK output<br><b>Don't change the bit. it is only used to debug.</b>                                                                                                                                                                                                                                                                                           | RW |
| 0 | MPLLEN | MPLL Enable. When MPLLEN is set to 1, MPLL starts to lock phase. After MPLL stabilizes, MPLL_ON bit is set.                                                                                                                                                                                                                                                                                         | RW |

### 18.1.2.29 EPLL Control Register

The EPLL Control Register (CPEPCR) is a 32-bit read/write register, which controls PLL multiplier, on/off state and stabilize time. It is initialized to 0x00000000 only by any reset. Only word access can be used on CPEPCR.

|               |       |    |    |    |    |    |    |    |    |    |    |    |    |       |    |                   |    |    |    |        |    |    |         |         |       |       |         |         |       |        |        |   |   |
|---------------|-------|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|-------------------|----|----|----|--------|----|----|---------|---------|-------|-------|---------|---------|-------|--------|--------|---|---|
| <b>CPEPCR</b> |       |    |    |    |    |    |    |    |    |    |    |    |    |       |    | <b>0x10000018</b> |    |    |    |        |    |    |         |         |       |       |         |         |       |        |        |   |   |
| <b>Bit</b>    | 31    | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18    | 17 | 16                | 15 | 14 | 13 | 12     | 11 | 10 | 9       | 8       | 7     | 6     | 5       | 4       | 3     | 2      | 1      | 0 |   |
|               | EPLLM |    |    |    |    |    |    |    |    |    |    |    |    | EPLLN |    |                   |    |    |    | EPLL0D |    |    | ERFSLIP | EFBSLIP | ELOCK | ETEST | EPLL_ON | EF_MODE | EBW_M | EPLLBP | EPLLEN |   |   |
| <b>RST</b>    | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0                 | 0  | 0  | 0  | 0      | 0  | 0  | 0       | 0       | 0     | 0     | 0       | 0       | 0     | 0      | 0      | 0 | 0 |

| Bits  | Name    | Description                                                                                                                                                                                      | RW |
|-------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:19 | EPLLM   | EPLL feedback 13-bit multiplication factor<br>$NF = EPLLM[12:0] + 1$                                                                                                                             | RW |
| 18:13 | EPLLN   | EPLL input reference 6-bit divider.<br>$NR = EPLLN[5:0] + 1$                                                                                                                                     | RW |
| 12:9  | EPLL0D  | EPLL post VCO divider<br>$OD = EPLL0D[3:0] + 1$ , this bit should always be odd number because vco doesn't support odd frequency divide. but OD can be 1.                                        | RW |
| 8     | ERFSLIP | ERFSLIP goes active for one or more divided feedback VCO cycles when the phase detector misses a divided reference cycle, i.e. when the VCO is running too fast. <b>It is only used to debug</b> | R  |
| 7     | EFBSLIP | EFBSLIP goes active for one or more divided reference cycles when the phase detector misses a divided feedback VCO cycle, i.e. when                                                              | R  |

|   |             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |    |
|---|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|   |             | the VCO is running too slow.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |
| 6 | ELOCK       | 0: the PLL output is not stable<br>1: the PLL output is stable<br>Software should clear this bit to 0, when this bit equal to 1, it indicates that PLL hadn't stable previously , <b>it is only used to debug.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | R  |
| 5 | ETEST       | <b>Don't change the bit</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW |
| 4 | EPLL_ON     | PLL ON Flag.<br>0: PLL is off or not stable<br>1: PLL is on and stable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | R  |
| 3 | EF_MOD<br>E | 0: usually frequency change mode<br>1: Synchronous frequency change mode<br>Such a counter change can be performed only if the initial counter values and the changed counter values lead to valid operating frequencies.<br>Otherwise, the PLL could lose lock. If the change in the VCO frequency is less than $Fbw\_3dB * NR / Fref$ (where $Fbw\_3db$ is the 3db bandwidth), the PLL will linearly settle (smoothly transition) to the new frequency. This constraint usually means that the frequency should be changed by less than 20%. Changes in excess of 20% can result in large frequency transients, which may force the PLL to lose lock. This state may necessitate putting the PLL through the reset sequence to re-achieve lock. Each linear settling change will require on the order of $1/Fbw$ (where $Fbw$ is the loop bandwidth), to mostly settle out. This approach is not suitable for simultaneous changes on PLLM and PLLN, as the approach does not guarantee that the PLLM and PLLN values will be observed on the same phase detection cycle | RW |
| 2 | EBW_M       | 0: BWADJ is controlled by hardware<br>1: BWADJ is controlled by software<br><b>Don't change the bit. it is only used to debug.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW |
| 1 | EPLLBP      | PLL Bypass.<br>0: PLL output<br>1: EXCLK output<br><b>Don't change the bit. it is only used to debug.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW |
| 0 | EPLLEN      | EPLL Enable. When EPLLEN is set to 1, EPLL starts to lock phase. After EPLL stabilizes, EPLL_ON bit is set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW |

### 18.1.2.30 VPLL Control Register

The VPLL Control Register (CPVPCR) is a 32-bit read/write register, which controls PLL multiplier, on/off state and stabilize time. It is initialized to 0x00000000 only by any reset. Only word access can be used on CPVPCR.

|               |                  |
|---------------|------------------|
| <b>CPVPCR</b> | <b>0x100001C</b> |
|---------------|------------------|

|     |       |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |        |    |    |         |         |       |       |         |         |       |        |        |   |   |   |
|-----|-------|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|--------|----|----|---------|---------|-------|-------|---------|---------|-------|--------|--------|---|---|---|
| Bit | 31    | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18    | 17 | 16 | 15 | 14 | 13 | 12     | 11 | 10 | 9       | 8       | 7     | 6     | 5       | 4       | 3     | 2      | 1      | 0 |   |   |
|     | VPLLM |    |    |    |    |    |    |    |    |    |    |    |    | VPLLN |    |    |    |    |    | VPLLOD |    |    | VRFSLIP | VFBSLIP | VLOCK | VTEST | VPLL_ON | VF_MODE | VBW_M | VPLLBP | VPLLEN |   |   |   |
| RST | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0       | 0       | 0     | 0     | 0       | 0       | 0     | 0      | 0      | 0 | 0 | 0 |

| Bits  | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW |
|-------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:19 | VPLLM       | VPLL feedback 13-bit multiplication factor<br>NF = VPLLM[12:0] + 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW |
| 18:13 | VPLLN       | VPLL input reference 6-bit divider.<br>NR = VPLLN[5:0] + 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RW |
| 12:9  | VPLLOD      | VPLL post VCO divider<br>OD = VPLLOD[3:0] + 1, this bit should always be odd number because vco doesn't support odd frequency divide. but OD can be 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW |
| 8     | VRFSLIP     | VRFSLIP goes active for one or more divided feedback VCO cycles when the phase detector misses a divided reference cycle, i.e. when the VCO is running too fast. <b>It is only used to debug</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | R  |
| 7     | VFBSLIP     | VFBSLIP goes active for one or more divided reference cycles when the phase detector misses a divided feedback VCO cycle, i.e. when the VCO is running too slow.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | R  |
| 6     | VLOCK       | 0: the PLL output is not stable<br>1: the PLL output is stable<br>Software should clear this bit to 0, when this bit equal to 1, it indicates that PLL hadn't stable previously , <b>it is only used to debug.</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | R  |
| 5     | VTEST       | <b>Don't change the bit</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW |
| 4     | VPLL_ON     | PLL ON Flag.<br>0: PLL is off or not stable<br>1: PLL is on and stable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | R  |
| 3     | VF_MOD<br>E | 0: usually frequency change mode<br>1: Synchronous frequency change mode<br>Such a counter change can be performed only if the initial counter values and the changed counter values lead to valid operating frequencies.<br>Otherwise, the PLL could lose lock. If the change in the VCO frequency is less than Fbw_3dB*NR/Fref (where Fbw_3db is the 3db bandwidth), the PLL will linearly settle (smoothly transition) to the new frequency. This constraint usually means that the frequency should be changed by less than 20%. Changes in excess of 20% can result in large frequency transients, which may force the PLL to lose lock. This state may necessitate putting the PLL through the reset sequence to re-achieve lock. Each linear settling change will require on the order of 1/Fbw (where Fbw is the loop bandwidth), to mostly settle out. | RW |

|   |        |                                                                                                                                                                                              |    |
|---|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|   |        | This approach is not suitable for simultaneous changes on PLLM and PLLN, as the approach does not guarantee that the PLLM and PLLN values will be observed on the same phase detection cycle |    |
| 2 | VBW_M  | 0: BWADJ is controlled by hardware<br>1: BWADJ is controlled by software<br><b>Don't change the bit.</b>                                                                                     | RW |
| 1 | VPLLBP | PLL Bypass.<br>0: PLL output<br>1: EXCLK output<br><b>Don't change the bit.</b>                                                                                                              | RW |
| 0 | VPLEN  | VPLL Enable. When VPLEN is set to 1, VPLL starts to lock phase.<br>After VPLL stabilizes, VPLL_ON bit is set.                                                                                | RW |

### 18.1.3 PLL Operation

The PLL developed as a macro cell for clock generator. It can generate a stable high-speed clock from a slower clock signal. The output frequency is adjustable and can be up to 2000MHz. The PLL integrates a phase frequency detector (PFD), a low pass filter (LPF), a voltage controlled oscillator (VCO) and other associated support circuitry. All fundamental building blocks as well as fully programmable dividers are integrated on the core. It is useful for clock multiplication of stable crystal oscillator sources and for de-skew clock signals.

The PLL block diagram is shown in following figure:

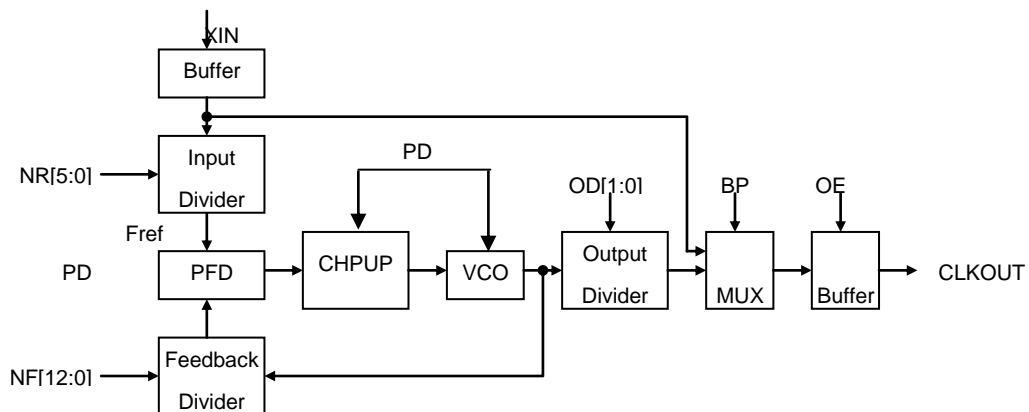


Figure 18-1 Block Diagram of PLL

#### 18.1.3.1 PLL Configuration

##### PLL Divider Value Setting

There are 3 divider values (N, M and NO) to set the PLL output clock frequency  $CLKOUT$ :

The PLL output frequency,  $F_{OUT}$ , is determined by the ratio set between the value set in the input divider and the feedback divider. PLL output frequency  $F_{OUT}$  is calculated from the following equations:

$$NF = PLLM + 1;$$

$$NR = PLLN + 1;$$

$$NO = PLLOD + 1; \text{ Note: } NO = 1, 2-16 \text{ (even only)}$$

$$FOUT = FIN * NF / NR / NO$$

$$FREF = FIN/NR \text{ range between } 183K - 1.5 \text{ GZ}$$

$$VCO = FIN * NF/NR \text{ range between } 300M - 1.5GZ \text{ and as large as possible.}$$

#### 18.1.4 Main Clock Division Change Sequence

Main clock (CCLK, L2CLK, H2CLK, PCLK) frequencies can be changed separately or simultaneously by changing division ratio. Following conditions must be obeyed:

- 1 the frequency of CCLK must be 1,2,3 or 4 times of the frequency of L2CLK
- 2 the frequency of H2CLK must be 1 or 2 times of the frequency of PCLK

**Don't violate this limitation, otherwise unpredictable error may occur.**

**Important Note: When cpu enters sleep mode, the apllen must be set to 1.**

## 18.2 Power Manager

In the Low-Power mode, part or whole processor is halted. This will reduce power consumption. The Power Management Controller contains low-power mode control and reset sequence control.

### 18.2.1 Low-Power Modes and Function

The processor supports six low-power modes and function:

- NORMAL mode  
In Normal mode, all peripherals and the basic blocks including power management block, the CPU core, the bus controller, the memory controller, the interrupt controller, DMA, and the external master may operate completely. But, the clock to each peripheral, except the basic blocks, can be stopped selectively by software to reduce the power consumption.
- DOZE mode  
DOZE mode is entered by setting DOZE bit of LCR to 1. In DOZE mode, clock is burst to CPU core and the clock duty is set by DUTY field of LCR. DOZE mode is canceled by reset, interrupt or clearing DOZE bit to 0. Continuous clock is supplied immediately after DOZE mode is canceled. The other Clocks except CCLK run continuously in DOZE mode.
- IDLE mode  
In IDLE mode, the clock to the CPU core is stopped except the bus controller, the memory controller, the interrupt controller, and the power management block. To exit the IDLE mode, the any interrupts should be activated.
- SLEEP mode



In SLEEP mode, all clocks except RTC clock are disabled. PLL is disabled also. SLEEP mode is canceled by reset or interrupt. When SLEEP mode is canceled, PLL is restarted, the PLL needs clock stabilization time (PLL lock time). This PLL stabilization time is automatically inserted by the internal logic with lock time count register. and all clocks start operating after PLL stability time.

- CLOCK GATE function**  
 CLOCK GATE function is used to gate specified on-chip module when it is not used. Set specified CLKGR0~40 bits in CLKGR will enter specified CLK gate function. CLOCK gate function is canceled by reset or clearing specified CLKGR0~40 to 0.
- Power down Mode**  
 In order to reduce power leakage, software may shut down power supply for AHB1 and GPS module. When system enters into SLEEP mode, the software may shut down power for J1 according to OPCR.PD bit.

### 18.2.2 Register Description

All PMC register 32bit access address is physical address.

**Table 18-2 Power/Reset Management Controller Registers Configuration**

| Name    | description                           | RW | Initial Value | Address    | Access Size |
|---------|---------------------------------------|----|---------------|------------|-------------|
| LCR     | Low Power Control Register            | RW | 0x000000F8    | 0x10000004 | 32          |
| SPCR0   | SRAM Power Control Register0          | RW | 0x00000000    | 0x100000B8 | 32          |
| SPCR1   | SRAM Power Control Register1          | RW | 0x00000000    | 0x100000BC | 32          |
| PSWC0ST | Power Switch Chain0 Start Time        | RW | 0x00000000    | 0x10000090 | 32          |
| PSWC1ST | Power Switch Chain1 Start Time        | RW | 0x00000000    | 0x10000094 | 32          |
| PSWC2ST | Power Switch Chain2 Start Time        | RW | 0x00000000    | 0x10000098 | 32          |
| PSWC3ST | Power Switch Chain3 Start Time        | RW | 0x00000000    | 0x1000009c | 32          |
| CLKGR0  | Clock Gate Register0                  | RW | 0xFFFFFFFFE0  | 0x10000020 | 32          |
| CLKGR1  | Clock Gate Register1                  | RW | 0xFFFFFFFF    | 0x10000028 | 32          |
| SRBC    | Soft Reset and Bus Control Register   | RW | 0x00000000    | 0x100000C4 | 32          |
| SLBC    | Sleep Boot Control Register           | RW | 0x00000000    | 0x100000C8 | 32          |
| SLPC    | Sleep PC Register                     | RW | 0x????????    | 0x100000CC | 32          |
| ERNG    | Enable rng regsiter                   | RW | 0x00000000    | 0x100000D8 | 32          |
| RNG     | rng regsiter                          | RW | 0x????????    | 0x100000DC | 32          |
| OPCR    | Oscillator and Power Control Register | RW | 0x000015C0    | 0x10000024 | 32          |

#### 18.2.2.1 Low Power Control Register

The Low Power Control Register (LCR) is a 32-bit read/write register that controls low-power mode

status. It is initialized to 0x000000F8 by any reset.

|     |         |        |        |        |       |      |      |      |          |    |    |          |     |    |    |    |            |    |    |    |    |    |      |   |   |   |   |      |     |   |   |   |
|-----|---------|--------|--------|--------|-------|------|------|------|----------|----|----|----------|-----|----|----|----|------------|----|----|----|----|----|------|---|---|---|---|------|-----|---|---|---|
|     | LCR     |        |        |        |       |      |      |      |          |    |    |          |     |    |    |    | 0x10000004 |    |    |    |    |    |      |   |   |   |   |      |     |   |   |   |
| Bit | 31      | 30     | 29     | 28     | 27    | 26   | 25   | 24   | 23       | 22 | 21 | 20       | 19  | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9    | 8 | 7 | 6 | 5 | 4    | 3   | 2 | 1 | 0 |
|     | PD_SCPU | PD_VPU | PD_GPU | PD_GPS | SCPUS | VPUS | GPUS | GPSS | Reserved |    |    | GPU_IDLE | PST |    |    |    |            |    |    |    |    |    | DUTY |   |   |   |   | DOZE | LPM |   |   |   |
| RST | 0       | 0      | 0      | 0      | 0     | 0    | 0    | 0    | 0        | 0  | 0  | 0        | 0   | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0    | 0 | 1 | 1 | 1 | 1    | 1   | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                             | RW |
|-------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | PD_SCPU  | Power Down Secondary CPU<br>0: not shut down power supply to Secondary CPU<br>1: shut down power supply to Secondary CPU                                                                                | RW |
| 30    | PD_VPU   | Power Down Module VPU.<br>0: not shut down power supply to VPU<br>1: shut down power supply to VPU                                                                                                      | RW |
| 29    | PD_GPU   | Power Down Module GPU.<br>0: not shut down power supply to GPU<br>1: shut down power supply to GPU                                                                                                      | RW |
| 28    | PD_GPS   | Power Down Module GPS.<br>0: not shut down power supply to GPS<br>1: shut down power supply to GPS                                                                                                      | RW |
| 27    | SCPUS    | Secondary CPU power down status.<br>0: Secondary CPU module not shut down<br>1: Secondary CPU module has entered shut down mode                                                                         | R  |
| 26    | VPUS     | VPU power down status.<br>0: VPU module not shut down<br>1: VPU module has entered shut down mode                                                                                                       | R  |
| 25    | GPUS     | GPU power down status.<br>0: GPU module not shut down<br>1: GPU module has entered shut down mode                                                                                                       | R  |
| 24    | GPSS     | GPS power down status.<br>0: GPS module not shut down<br>1: GPS module has entered shut down mode                                                                                                       | R  |
| 23:21 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                    | R  |
| 20    | GPU_IDLE | When GPU_IDLE high indicates that the GPU core is idle can be powered off.                                                                                                                              | R  |
| 19:8  | PST      | Power stability Time. Specifies the Power stabilize time by unit of RTCCLK (approximate 32kHz) cycles.                                                                                                  | RW |
| 7:3   | DUTY     | CPU Clock Duty. Control the CPU clock duty in doze mode. When the DUTY field is 0x1F, the clock is always on and when it is zero, the clock is always off. Set the DUTY field to 0 when the CPU will be | RW |

|     |      |                                                                                                                                                                                                                                                                                                                         |    |
|-----|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|     |      | disabled for an extended amount of time.<br>00000: 0/31 duty-cycle<br>00001: 1/31 duty-cycle<br>00010: 2/31 duty-cycle<br>...<br>11111: 31/31 duty-cycle                                                                                                                                                                |    |
| 2   | DOZE | Doze Mode. Control the doze mode. When doze mode is canceled, this bit is cleared to 0 automatically.<br>0: Doze mode is off<br>1: Doze mode is on                                                                                                                                                                      | RW |
| 1:0 | LPM  | Low Power Mode. Specifies which low-power mode will be entered when SLEEP instruction is executed.<br>Bit 1~0:<br><b>00</b> : IDLE mode will be entered when SLEEP instruction is executed<br><b>01</b> : SLEEP mode will be entered when SLEEP instruction is executed<br><b>10</b> : Reserved<br><b>11</b> : Reserved | RW |

### 18.2.2.2 SRAM Power Control Register0

The SRAM Power Control Register0 (SPCR0) is a 32-bit read/write register that controls SRAM power status. It is initialized to 0x00000000 by any reset.

| SPCR0 |      |     |      |      |     |       |       |       |       |       |      |     |     |     |      | 0x100000B8 |       |       |      |      |      |     |     |     |       |         |     |     |     |      |   |   |
|-------|------|-----|------|------|-----|-------|-------|-------|-------|-------|------|-----|-----|-----|------|------------|-------|-------|------|------|------|-----|-----|-----|-------|---------|-----|-----|-----|------|---|---|
| Bit   | 31   | 30  | 29   | 28   | 27  | 26    | 25    | 24    | 23    | 22    | 21   | 20  | 19  | 18  | 17   | 16         | 15    | 14    | 13   | 12   | 11   | 10  | 9   | 8   | 7     | 6       | 5   | 4   | 3   | 2    | 1 | 0 |
|       | SSI1 | PCM | I2S1 | I2S0 | TCU | UART4 | UART3 | UART2 | UART1 | UART0 | PDMA | GPS | UHC | MAC | MSC2 | MSC1       | TSSI1 | TSSI0 | OTG1 | OTG0 | HDMI | LCD | IPU | CIM | GPVLC | COMPRES | X2D | DES | SCC | SADC |   |   |
| RST   | 0    | 0   | 0    | 0    | 0   | 0     | 0     | 0     | 0     | 0     | 0    | 0   | 0   | 0   | 0    | 0          | 0     | 0     | 0    | 0    | 0    | 0   | 0   | 0   | 0     | 0       | 0   | 0   | 0   | 0    | 0 | 0 |

| Bits | Name | Description                                           | RW |
|------|------|-------------------------------------------------------|----|
| 31   | SSI1 | Memory Power control for SSI1<br>0: on<br>1: shutdown | RW |
| 30   | PCM  | Memory Power control for CPM<br>0: on<br>1: shutdown  | RW |
| 29   | I2S1 | Memory Power control for I2S1<br>0: on<br>1: shutdown | RW |
| 28   | I2S0 | Memory Power control for I2S0                         | RW |

|    |       |                                                        |    |
|----|-------|--------------------------------------------------------|----|
|    |       | 0: on<br>1: shutdown                                   |    |
| 27 | TCU   | Memory Power control for TCU<br>0: on<br>1: shutdown   | RW |
| 26 | UART4 | Memory Power control for UART4<br>0: on<br>1: shutdown | RW |
| 25 | UART3 | Memory Power control for UART3<br>0: on<br>1: shutdown | RW |
| 24 | UART2 | Memory Power control for UART2<br>0: on<br>1: shutdown | RW |
| 23 | UART1 | Memory Power control for UART1<br>0: on<br>1: shutdown | RW |
| 22 | UART0 | Memory Power control for UART0<br>0: on<br>1: shutdown | RW |
| 21 | PDMA  | Memory Power control for PDMA<br>0: on<br>1: shutdown  | RW |
| 20 | GPS   | Memory Power control for GPS<br>0: on<br>1: shutdown   | RW |
| 19 | UHC   | Memory Power control for UHC<br>0: on<br>1: shutdown   | RW |
| 18 | MAC   | Memory Power control for MAC<br>0: on<br>1: shutdown   | RW |
| 17 | MSC2  | Memory Power control for MSC2<br>0: on<br>1: shutdown  | RW |
| 16 | MSC1  | Memory Power control for MSC1<br>0: on<br>1: shutdown  | RW |
| 15 | TSSI1 | Memory Power control for TSSI1<br>0: on<br>1: shutdown | RW |
| 14 | TSSI0 | Memory Power control for TSSI0<br>0: on                | RW |

|     |          |                                                           |    |
|-----|----------|-----------------------------------------------------------|----|
|     |          | 1: shutdown                                               |    |
| 13  | OTG1     | Memory Power control for OTG1<br>0: on<br>1: shutdown     | RW |
| 12  | OTG0     | Memory Power control for OTG0<br>0: on<br>1: shutdown     | RW |
| 11  | HDMI     | Memory Power control for HDMI<br>0: on<br>1: shutdown     | RW |
| 10  | LCD      | Memory Power control for LCD<br>0: on<br>1: shutdown      | RW |
| 9   | IPU      | Memory Power control for IPU<br>0: on<br>1: shutdown      | RW |
| 8   | CIM      | Memory Power control for CIM<br>0: on<br>1: shutdown      | RW |
| 7   | GPVLC    | Memory Power control for GPVLC<br>0: on<br>1: shutdown    | RW |
| 6   | COMPRESS | Memory Power control for COMPRESS<br>0: on<br>1: shutdown | RW |
| 5   | X2D      | Memory Power control for X2D<br>0: on<br>1: shutdown      | RW |
| 4   | DES      | Memory Power control for DES<br>0: on<br>1: shutdown      | RW |
| 3   | SCC      | Memory Power control for SCC<br>0: on<br>1: shutdown      | RW |
| 2   | SADC     | Memory Power control for SADC<br>0: on<br>1: shutdown     | RW |
| 1:0 | Reserved |                                                           | R  |

**Important Note :**

**When shutting down the power of more than one ram, the corresponding controlling bits can be set to 1 at the same time.**

**But, when supplying power to more than one ram, these bits should be set to 0 sequentially.**

It will take at least 1us before ram can work from power off to power on

### 18.2.2.3 SRAM Power Control Register1

The SRAM Power Control Register0 (SPCR1) is a 32-bit read/write register that controls SRAM power status. It is initialized to 0x00000000 by any reset.

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |      |      |      |      |     |      |   |   |   |   |   |   |   |   |   |   |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|------|------|------|------|-----|------|---|---|---|---|---|---|---|---|---|---|
| SPCR1 |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x100000BC |      |      |      |      |     |      |   |   |   |   |   |   |   |   |   |   |
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15   | 14   | 13   | 12   | 11  | 10   | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | DDR1 | DDR0 | SSI0 | MSC0 | BCH | NEMC |   |   |   |   |   |   |   |   |   |   |
| RST   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0    | 0    | 0    | 0    | 0   | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                           | RW |
|------|----------|-------------------------------------------------------|----|
| 31:6 | Reserved |                                                       | R  |
| 5    | DDR1     | Memory Power control for DDR1<br>0: on<br>1: shutdown | RW |
| 4    | DDR0     | Memory Power control for DDR0<br>0: on<br>1: shutdown | RW |
| 3    | SSI0     | Memory Power control for SSI0<br>0: on<br>1: shutdown | RW |
| 2    | MSC0     | Memory Power control for MSC0<br>0: on<br>1: shutdown | RW |
| 1    | BCH      | Memory Power control for BCH<br>0: on<br>1: shutdown  | RW |
| 0    | NEMC     | Memory Power control for NEMC<br>0: on<br>1: shutdown | RW |

**Important Note :**

Software should take care when shutting down these rams of the above modules, especially from sleep mode restore.

### 18.2.2.4 SRAM Speed Control Register

The SRAM Speed Control Register (SSPC) is a 32-bit read/write register that controls SRAM Speed .

It is initialized to 0x00000000 by any reset.

|             |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |                   |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------------|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|-------------------|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>SSPC</b> |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    | <b>0x100000C0</b> |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>  | 31 | 30 | 29 | 28 | 27       | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             | MC |    |    |    | Reserved |    |    |    |    |    |    |    |    |    |    |                   | RME | RM |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bits  | Name     | Description                                                                   | RW |
|-------|----------|-------------------------------------------------------------------------------|----|
| 31:29 | MC       | CPU SRAM timing control.<br>Don't change the bit.<br>It is only used to debug | R  |
| 28:5  | Reserved |                                                                               | R  |
| 4     | RME      | Don't change the bit.<br>It is only used to debug                             | RW |
| 3:0   | RM       | Don't change the bit.<br>It is only used to debug                             | RW |

### 18.2.2.5 Power Switch Chain0 Start Time Register

|                |          |    |    |    |    |    |    |    |    |    |    |    |         |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|----------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| <b>PSWC0ST</b> |          |    |    |    |    |    |    |    |    |    |    |    |         |    |    | <b>0x10000090</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>     | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19      | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|                | Reserved |    |    |    |    |    |    |    |    |    |    |    | PSWC0ST |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>     | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0                 | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 18.2.2.6 Power Switch Chain1 Start Time Register

|                |          |    |    |    |    |    |    |    |    |    |    |    |         |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|----------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| <b>PSWC1ST</b> |          |    |    |    |    |    |    |    |    |    |    |    |         |    |    | <b>0x10000094</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>     | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19      | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|                | Reserved |    |    |    |    |    |    |    |    |    |    |    | PSWC1ST |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>     | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0                 | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 18.2.2.7 Power Switch Chain2 Start Time Register

|                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| <b>PSWC2ST</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x10000098</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|

|     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|-----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| Bit | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|     | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PSWC2ST |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 18.2.2.8 Power Switch Chain3 Start Time Register

|     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|     | PSWC3ST  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x1000009C |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|     | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PSWC3ST    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**NOTE:** The Start Time by the unit of PCLK cycles.

### 18.2.2.9 Clock Gate Register0

The Clock Gate Register (CLKGR0) is a 32-bit read/write register that controls the CLOCK GATE function of peripherals. It is reset to 0xFFFFF0.

|     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-----|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|---|
|     | CLKGR  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10000020 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Bit | 31     | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |   |
|     | CLKGR0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |
| RST | 0      | 0  | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1          | 1  | 1  | 1  | 1  | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bits | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW  |        |             |    |      |  |    |      |  |    |     |                                           |    |     |                                           |    |     |                                           |    |     |                                           |  |
|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|--------|-------------|----|------|--|----|------|--|----|-----|-------------------------------------------|----|-----|-------------------------------------------|----|-----|-------------------------------------------|----|-----|-------------------------------------------|--|
| 31:0 | CLKGR0 | Clock gate Bits. Controls the clock supplies to some peripherals. If set, clock supplies to associated devices are stopped, and registers of the device cannot be accessed also.                                                                                                                                                                                                                                                                                                                                                                                                     | RW  |        |             |    |      |  |    |      |  |    |     |                                           |    |     |                                           |    |     |                                           |    |     |                                           |  |
|      |        | <table border="1"> <thead> <tr> <th>Bit</th> <th>Module</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>31</td> <td>DDR1</td> <td></td> </tr> <tr> <td>30</td> <td>DDR0</td> <td></td> </tr> <tr> <td>29</td> <td>IPU</td> <td>After reset period, the clock is stopped.</td> </tr> <tr> <td>28</td> <td>LCD</td> <td>After reset period, the clock is stopped.</td> </tr> <tr> <td>27</td> <td>TVE</td> <td>After reset period, the clock is stopped.</td> </tr> <tr> <td>26</td> <td>CIM</td> <td>After reset period, the clock is stopped.</td> </tr> </tbody> </table> | Bit | Module | Description | 31 | DDR1 |  | 30 | DDR0 |  | 29 | IPU | After reset period, the clock is stopped. | 28 | LCD | After reset period, the clock is stopped. | 27 | TVE | After reset period, the clock is stopped. | 26 | CIM | After reset period, the clock is stopped. |  |
| Bit  | Module | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |     |        |             |    |      |  |    |      |  |    |     |                                           |    |     |                                           |    |     |                                           |    |     |                                           |  |
| 31   | DDR1   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |     |        |             |    |      |  |    |      |  |    |     |                                           |    |     |                                           |    |     |                                           |    |     |                                           |  |
| 30   | DDR0   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |     |        |             |    |      |  |    |      |  |    |     |                                           |    |     |                                           |    |     |                                           |    |     |                                           |  |
| 29   | IPU    | After reset period, the clock is stopped.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |     |        |             |    |      |  |    |      |  |    |     |                                           |    |     |                                           |    |     |                                           |    |     |                                           |  |
| 28   | LCD    | After reset period, the clock is stopped.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |     |        |             |    |      |  |    |      |  |    |     |                                           |    |     |                                           |    |     |                                           |    |     |                                           |  |
| 27   | TVE    | After reset period, the clock is stopped.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |     |        |             |    |      |  |    |      |  |    |     |                                           |    |     |                                           |    |     |                                           |    |     |                                           |  |
| 26   | CIM    | After reset period, the clock is stopped.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |     |        |             |    |      |  |    |      |  |    |     |                                           |    |     |                                           |    |     |                                           |    |     |                                           |  |



|  |  |    |       |                                           |  |
|--|--|----|-------|-------------------------------------------|--|
|  |  | 25 | SMB2  | After reset period, the clock is stopped. |  |
|  |  | 24 | UHC   | After reset period, the clock is stopped. |  |
|  |  | 23 | MAC   | After reset period, the clock is stopped. |  |
|  |  | 22 | GPS   | After reset period, the clock is stopped. |  |
|  |  | 21 | PDMA  |                                           |  |
|  |  | 20 | SSI2  | After reset period, the clock is stopped. |  |
|  |  | 19 | SSI1  | After reset period, the clock is stopped. |  |
|  |  | 18 | UART3 | After reset period, the clock is stopped. |  |
|  |  | 17 | UART2 | After reset period, the clock is stopped. |  |
|  |  | 16 | UART1 | After reset period, the clock is stopped. |  |
|  |  | 15 | UART0 | After reset period, the clock is stopped. |  |
|  |  | 14 | SADC  | After reset period, the clock is stopped. |  |
|  |  | 13 | KBC   | After reset period, the clock is stopped. |  |
|  |  | 12 | MSC2  | After reset period, the clock is stopped. |  |
|  |  | 11 | MSC1  | After reset period, the clock is stopped. |  |
|  |  | 10 | OWI   | After reset period, the clock is stopped. |  |
|  |  | 9  | TSSIO | After reset period, the clock is stopped. |  |
|  |  | 8  | AIC   | After reset period, the clock is stopped. |  |
|  |  | 7  | SCC   | After reset period, the clock is stopped. |  |
|  |  | 6  | SMB1  | After reset period, the clock is stopped. |  |
|  |  | 5  | SMB0  | After reset period, the clock is stopped. |  |
|  |  | 4  | SSI0  |                                           |  |
|  |  | 3  | MSC0  |                                           |  |
|  |  | 2  | OTG0  |                                           |  |
|  |  | 1  | BCH   |                                           |  |
|  |  | 0  | NEMC  |                                           |  |

### 18.2.2.10 Clock Gate Register1

The Clock Gate Register (CLKGR1) is a 32-bit read/write register that controls the CLOCK GATE function of peripherals. It is reset to 0x0000FFFF.

|               |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                  |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|---------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>CLKGR1</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x1000028</b> |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16               | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|               | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                  | CLKGR1 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                | 1      | 1  | 1  | 1  | 1  | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bits  | Name     | Description                          | RW |
|-------|----------|--------------------------------------|----|
| 31:12 | Reserved | Writing has no effect, read as zero. | R  |

| 11:0 | CLKGR1 | Clock gate Bits. Controls the clock supplies to some peripherals. If set, clock supplies to associated devices are stopped, and registers of the device cannot be accessed also. |              |                                           |
|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-------------------------------------------|
|      |        | Bit                                                                                                                                                                              | Module       | Description                               |
|      |        | 15                                                                                                                                                                               | P1           |                                           |
|      |        | 14                                                                                                                                                                               | X2D          | After reset period, the clock is stopped  |
|      |        | 13                                                                                                                                                                               | DES          |                                           |
|      |        | 12                                                                                                                                                                               | SMB4         | After reset period, the clock is stopped  |
|      |        | 11                                                                                                                                                                               | AHB_MON      | After reset period, the clock is stopped. |
|      |        | 10                                                                                                                                                                               | UART4        | After reset period, the clock is stopped. |
|      |        | 9                                                                                                                                                                                | HDMI         | After reset period, the clock is stopped. |
|      |        | 8                                                                                                                                                                                | OTG1         | After reset period, the clock is stopped. |
|      |        | 7                                                                                                                                                                                | GPVLC        | After reset period, the clock is stopped. |
|      |        | 6                                                                                                                                                                                | AIC1         | After reset period, the clock is stopped. |
|      |        | 5                                                                                                                                                                                | COMPRES<br>S | After reset period, the clock is stopped. |
|      |        | 4                                                                                                                                                                                | GPU          | After reset period, the clock is stopped. |
|      |        | 3                                                                                                                                                                                | PCM          | After reset period, the clock is stopped. |
| 2    | VPU    | After reset period, the clock is stopped.                                                                                                                                        |              |                                           |
| 1    | TSSI1  | After reset period, the clock is stopped.                                                                                                                                        |              |                                           |
| 0    | SMB3   | After reset period, the clock is stopped.                                                                                                                                        |              |                                           |

### 18.2.2.11 Soft Reset and Bus Control Register (SRBC)

The Soft Reset and Bus Control Register is a 32-bit read/write register that control some module reset and bus transmission. It is initialized to 0x0000000 0by reset.

| SRBC |        |         |         |          |        |         |         |            |    |        |         |         |    |        |         | 0x100000C4 |    |        |         |         |    |          |   |          |          |          |          |          |          |         |          |          |
|------|--------|---------|---------|----------|--------|---------|---------|------------|----|--------|---------|---------|----|--------|---------|------------|----|--------|---------|---------|----|----------|---|----------|----------|----------|----------|----------|----------|---------|----------|----------|
| Bit  | 31     | 30      | 29      | 28       | 27     | 26      | 25      | 24         | 23 | 22     | 21      | 20      | 19 | 18     | 17      | 16         | 15 | 14     | 13      | 12      | 11 | 10       | 9 | 8        | 7        | 6        | 5        | 4        | 3        | 2       | 1        | 0        |
|      | VPU_SR | VPU_STP | VPU_ACK | Reserved | GPU_SR | GPU_STP | GPU_ACK | GPU_ACTIVE |    | LCD_SR | LCD_STP | LCD_ACK |    | CIM_SR | CIM_STP | CIM_ACK    |    | UHC_SR | OTG1_SR | OTG0_SR |    | Reserved |   | AHB0_STP | AHB0_ACK | Reserved | AHB2_STP | AHB2_ACK | Reserved | DDR_STP | DDR1_ACK | DDR0_ACK |
| RST  | 0      | 0       | 0       | 0        | 0      | 0       | 0       | 0          | 0  | 0      | 0       | 0       | 0  | 0      | 0       | 0          | 0  | 0      | 0       | 0       | 0  | 0        | 0 | 0        | 0        | 0        | 0        | 0        | 0        | 0       | 0        | 0        |

| Bits | Name     | Description                                                            | RW |
|------|----------|------------------------------------------------------------------------|----|
| 31   | VPU_SR   | 0: VPU does not enter soft reset mode<br>1: VPU enters soft reset mode | RW |
| 30   | VPU_STP  | Request for VPU to Stop bus transfer.                                  | RW |
| 29   | VPU_ACK  | VPU Stop Ack.                                                          | R  |
| 28   | Reserved | Writing has no effect, read as zero.                                   | R  |
| 27   | GPU_SR   | 0: GPU does not enter soft reset mode                                  | RW |

|      |            |                                                                                |    |
|------|------------|--------------------------------------------------------------------------------|----|
|      |            | 1: GPU enters soft reset mode                                                  |    |
| 26   | GPU_STP    | Request for GPU to Stop bus transfer.                                          | RW |
| 25   | GPU_ACK    | GPU Stop Ack.                                                                  | R  |
| 24   | GPU_ACTIVE | GPU active<br>When send GPU_STP, software should check GPU_ACK and GPU_ACTIVE. | R  |
| 23   | Reserved   |                                                                                |    |
| 22   | LCD_SR     | 0: LCD does not enter soft reset mode<br>1: LCD enters soft reset mode         | RW |
| 21   | LCD_STP    | Request for LCD to Stop bus transfer.                                          | RW |
| 20   | LCD_ACK    | LCD Stop Ack.                                                                  | R  |
| 19   | Reserved   |                                                                                |    |
| 18   | CIM_SR     | 0: CIM does not enter soft reset mode<br>1: CIM enters soft reset mode         | RW |
| 17   | CIM_STP    | Request for CIM to Stop bus transfer.                                          | RW |
| 16   | CIM_ACK    | CIM Stop Ack.                                                                  | R  |
| 15   | Reserved   |                                                                                |    |
| 14   | UHC_SR     | UHC soft reset                                                                 | RW |
| 13   | OTG1_SR    | OTG1 soft reset                                                                | RW |
| 12   | OTG0_SR    | OTG0 soft reset                                                                | RW |
| 11:9 | Reserved   |                                                                                |    |
| 8    | AHB0_STP   | Request for AHB0 to Stop bus transfer.                                         | RW |
| 7    | AHB0_ACK   | AHB0 Stop Ack.                                                                 | R  |
| 6    | Reserved   |                                                                                |    |
| 5    | AHB2_STP   | Request for AHB2 to Stop bus transfer.                                         | RW |
| 4    | AHB2_ACK   | AHB2 Stop Ack.                                                                 | R  |
| 3    | Reserved   |                                                                                | R  |
| 2    | DDR0_STP   | Request for DDR to Stop bus transfer.                                          | RW |
| 1    | DDR1_ACK   | DDR Stop Ack.                                                                  |    |
| 0    | DDR0_ACK   | DDR Stop Ack.                                                                  | R  |

### 18.2.2.12 Sleep Boot Control Register (SLBC)

The Sleep Boot Control Register is a 32-bit read/write register that control sleep boot mode. It is initialized to 0x00000000 0by reset.

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                  |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|            | <b>SLBC</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x10000C8</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16               | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|     |          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |      |   |   |
|-----|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------|---|---|
|     | Reserved |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | SLBC |   |   |
| RST | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                       | RW |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:1 | Reserved |                                                                                                                                                                                                   | R  |
| 0    | SLBC     | 0: Sleep Mode Wake up Boot Process is same with any reset.<br>1: Sleep Mode Wake up Boot Process is jumped to SLPC, don't care Boot_Sel.<br>The bit is only valid When shut down P0 in Sleep Mode | RW |

### 18.2.2.13 Sleep PC Register (SLPC)

The Sleep PC Register is a 32-bit read/write register that control sleep mode jump address.

|      |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| SLPC |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x100000CC |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| Bit  | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|      | SLPC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST  | ?    | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?          | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description                                                                                                                        | RW |
|------|------|------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | SLPC | When SLBC is 1, sleep boot is jumped to SLPC, not true boot as any reset.<br>The bit is only valid When shut down P0 in Sleep Mode | RW |

### 18.2.2.14 Enable Rng Register (ERNG)

The Enable rng Register is a 32-bit read/write register that control whether enable rng register. It is initialized to 0x00000000 by reset.

|      |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
|------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------|
| ERNG |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x100000D8 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
| Bit  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |      |
|      | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | ERNG |
| RST  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    |

| Bits | Name | Description | RW |
|------|------|-------------|----|
|------|------|-------------|----|

|      |          |                            |    |
|------|----------|----------------------------|----|
| 31:1 | Reserved |                            | R  |
| 0    | ERNG     | 0: not enable<br>1: enable | RW |

### 18.2.2.15 Rng Register (RNG)

The rng Register is a 32-bit read/write registers.

|            |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|------------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|
|            | <b>RNG</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x100000DC</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31         | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |
|            | RNG        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b> | ?          | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                 | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description | RW |
|------|------|-------------|----|
| 31:0 | RNG  |             | RW |

### 18.2.2.16 Oscillator and Power Control Register (OPCR)

The Oscillator and Power Control Register is a 32-bit read/write register that specifies some special controls to oscillator and analog block. It is initialized to 0x000015C0 by reset.

|            |             |           |         |    |         |        |          |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |   |   |   |   |         |         |          |      |    |      |          |     |
|------------|-------------|-----------|---------|----|---------|--------|----------|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|---|---|---|---|---------|---------|----------|------|----|------|----------|-----|
|            | <b>OPCR</b> |           |         |    |         |        |          |    |    |    |    |    |    |    |    |    | <b>0x10000024</b> |    |    |    |    |    |   |   |   |   |         |         |          |      |    |      |          |     |
| <b>Bit</b> | 31          | 30        | 29      | 28 | 27      | 26     | 25       | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5       | 4       | 3        | 2    | 1  | 0    |          |     |
|            | IDLE_DIS    | GPU_CLKEN | MASKVOU |    | L2CM_PD | RST_EN | Reserved |    |    |    |    |    |    |    |    |    | O1ST              |    |    |    |    |    |   |   |   |   | SPENDNO | SPENDN1 | BUS_MODE | O1SE | PD | ERCS | CPU_MODE | OSE |
| <b>RST</b> | 0           | 0         | 0       | 0  | 0       | 0      | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0  | 0  | 0  | 1  | 0  | 1 | 0 | 1 | 1 | 1       | 0       | 0        | 0    | 0  | 0    | 0        |     |

| Bits  | Name      | Description                                                                                                                                                                 | RW |
|-------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | IDLE_DIS  | 0: when CPU enters idle mode, CPU clock is stopped<br>1: When CPU enters idle mode, CPU clock is not stopped                                                                | RW |
| 30    | GPU_CLKEN | 0: GPU CLK is running<br>1: GPU CLK is stopped                                                                                                                              | RW |
| 29    | MASK_VPU  | Don't change this bit , it is only used to debug                                                                                                                            | RW |
| 28    | Reserved  |                                                                                                                                                                             |    |
| 27:26 | L2CM_PD   | 00 : L2CACHE memory doesn't power down in sleep mode<br>01 : L2CACHE memory enters retention mode in sleep mode<br>10 : L2CACHE memory enters power down mode in sleep mode | RW |

|       |          |                                                                                                                                                                                                           |    |
|-------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 25    | RST_EN   | <b>Should be 0. software must not write the bit, otherwise the error occurred</b>                                                                                                                         | RW |
| 24:16 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                      | R  |
| 15:8  | O1ST     | EXCLK Oscillator Stabilize Time. This filed specifies the EXCLKoscillator stabilize time by unit of 16 RTCCLK periods (oscillator stable time O1ST × 16 / 32768) cycles. It is initialized to H'15.       | RW |
| 7     | SPENDN0  | force port0(otg) to enter suspend mode.<br>0: port0(otg) has forced to entered SUSPEND mode<br>1: port0(otg) hasn't forced to entered SUSPEND mode                                                        | RW |
| 6     | SPENDN1  | force port1(uhc) to enter suspend mode.<br>0: port1(uhc) has forced to entered SUSPEND mode<br>1: port1(uhc) hasn't forced to entered SUSPEND mode                                                        | RW |
| 5     | BUS_MODE | The default value is 0 . when the value is 1, the bus access is bursted                                                                                                                                   | RW |
| 4     | O1SE     | EXCLK Oscillator Sleep Mode Enable. This filed controls the state of the EXCLK oscillator in Sleep mode.<br>0: EXCLK oscillator is disabled in Sleep mode<br>1: EXCLK oscillator is enabled in Sleep mode | RW |
| 3     | PD       | The fief controls the state P0 in Sleep mode.<br>0: The P0 not power down in Sleep mode<br>1: The P0 power down in Sleep mode                                                                             | RW |
| 2     | ERCS     | EXCLK/512 clock and RTCLK clock selection.<br>0: select EXCLK/512 division ration clock<br>1: select RTCLK clock<br>the clock only output to CPM INTC SSI TCU etc.                                        | RW |
| 1     | CPU_MODE | The default value is 0 When the value is 1, the CPU access is acclerated.                                                                                                                                 | RW |
| 0     | OSE      | 0: EXCLK enable<br>1: EXCLK disable<br>When SRC_SEL == 11 of CPCCR, the software may set this bit.                                                                                                        | RW |

### 18.2.3 Doze Mode

Firstly, software should set the DUTY bits of LCR. Then set DOZE bit of LCR to 1 to enter doze mode. When slot controller of PMC indicates that the CPU clock's time-slot has expired, CPU is halted but its register contents are retained. During doze mode, program can modify clock duty-cycle according to core resource requirement. Clock control is in increments of approximately 3% (1/31).

Doze is exited by software, interrupt, reset or SLEEP instruction.

### 18.2.4 IDLE Mode

In normal mode, when LPM bits in LCR are 0 and SLEEP instruction is executed, the processor enters idle mode. CPU is halted but its register contents are retained. All critical application must be finished and peripherals must be configured to generate interrupts when they need CPU attention.

The procedure of entering sleep mode is shown below:

- 1 Set LPM bits in LCR to 0.
- 2 Executes SLEEP instruction.
- 3 When current operation of CPU core has finished and CPU core is idle, CCLK supply to CPU core is stopped.

IDLE mode is exited by an interrupt (IRQ or on-chip devices) or a reset.

### 18.2.5 SLEEP Mode

In normal mode, when LPM bits in LCR is 1 and SLEEP instruction is executed, the processor enters SLEEP mode. CPU and on-chip devices are halted, except some wakeup-logic. PLL is shut off. Clock output from CKO pin is also stopped. SDRAM content is preserved by driving into self-refresh state. CPU registers and on-chip devices registers contents are retained.

Before entering SLEEP mode, software should ensure that all peripherals are not running. The procedure of entering SLEEP mode is shown below:

- 1 Set LPM bit in LCR to 1.
- 2 Execute a SLEEP instruction.
- 3 When current access on system bus complete, the arbiter will not grant any following request. EMC will drive SDRAM from auto-refresh mode to self-refresh mode.
- 4 When system bus is idle state and SDRAM is self-refresh mode, internal clock supplies are stopped.
- 5 SLEEP mode can be exited by an interrupt (IRQ or on-chip devices), WDT reset or a power-on reset via the RESET pin.

### 18.2.6 Power Down Mode

When PD\_VPU bit in LCR is 1, the processor enters shut down VPU module power sequence.

When PD\_VPUS bit in LCR is 1, it indicates that the VPU module has been shut off. The leakage current of VPU is reduced almost to 0.

When entering sleep mode, when PD bits in OPCR is 1, the J1 power supply would be shut off. The leakage current of J1 is reduced almost to 0.

The procedure of entering Power Down mode is shown below:

- 1 set proper values for PSWC0ST, PSWC1ST, PSWC2ST, PSWC3ST.
- 2 set PD\_VPU bit in LCR to 1.

- 3 wait until PD\_VPUS = 1.
- 4 When need for supply power for VPU, set PD\_VPU in LCR to 0.
- 5 wait until PD\_VPUS = 0.
- 6 the hardware auto generate RESET signal to the module, the software must again config the module as the same to POWER ON Reset.

The same as GPU , GPS, P1, software should power on these modules one by one.

### 18.3 Reset Control Module

#### 18.3.1 Register Description

All RCM register 32bit access address is physical address.

| Name | description           | RW | Initial Value | Address    | Access Size |
|------|-----------------------|----|---------------|------------|-------------|
| RSR  | Reset Status Register | RW | 0x????????    | 0x10000008 | 32          |

##### 18.3.1.1 Reset Status Register (RSR)

The Reset Status Register (RSR) is a 32-bit read/write register which records last cause of reset. Each RSR bit is set by a different source of reset. Please refer to Reset Sequence Control for reset sources description.

| RSR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10000008 |    |     |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|-----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14  | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|     | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | HR | POR | WR | PR |    |    |   |   |   |   |   |   |   |   |   |   |
| RST | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0   | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ? | ? | ? |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                           | RW |
|------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31   | RETEN    | 0 : hibernate or power on reset without retention mode<br>1: hibernate or power on reset without retention mode                                                                                                                                                                                                                                       | R  |
| 30:4 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                  | R  |
| 3    | HR       | Hibernate Reset.                                                                                                                                                                                                                                                                                                                                      |    |
| 2    | POR      | P0 power up Reset. It indicates that P0 has been shut down, now it has been power up. When P0 reset is detected, POR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored.<br><br>0: P0 reset has not occurred since the last time the software clears this bit | RW |



|   |    |                                                                                                                                                                                                                                                                                                                                                                                      |    |
|---|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|   |    | 1: P0 reset has occurred since the last time the software clears this bit                                                                                                                                                                                                                                                                                                            |    |
| 1 | WR | WDT Reset. When a WDT reset is detected, WR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored.<br>0: WDT reset has not occurred since the last time the software clears this bit<br>1: WDT reset has occurred since the last time the software clears this bit                              | RW |
| 0 | PR | Power On Reset. When a poweron reset via PRESET pin is detected, PR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 is ignored.<br>0: Power on reset has not occurred since the last time the software clears this bit<br>1: Power on reset has occurred since the last time the software clears this bit | RW |

### 18.3.2 Power On Reset

Power on reset is generated when PRESET pin is driven to low. Internal reset is asserted immediately. All pins return to their reset states. The Power on reset is extended to 40MS.

PRESET pin must be held low until power stabilizes and the EXCLK oscillator stabilize. CPU and peripherals are clocked by EXCLK oscillator output directly. PLL is reset to off state. All internal modules are initialized to their predefined reset states.

### 18.3.3 WDT Reset

WDT reset is generated when WDT overflow. Internal reset is asserted within two RTCCLK cycles. All pins return to their reset states.

Then WDT reset source is cleared because of internal reset. The internal reset is asserted for about 10 milliseconds. CPU and peripherals are clocked by EXCLK oscillator output directly. PLL is reset to off state.

## 19 Timer/Counter Unit

### 19.1 Overview

The TCU (Timer/Counter with PWM output) contains 8 channels of 16-bit programmable timers (timers 0 to 5). They can be used as Timer or PWM.

TCU has the following features:

- There are two modes of TCU for the eight channels
  - TCU1: Channel 0, 3,4, 5, 6,and 7
  - TCU2: Channel 1,2
- Six independent channels, each consisting of
  - Counter
  - Data register (FULL and HALF)
  - Control register
- Independent clock for each counter, selectable by software
  - PCLK, EXTAL and RTCCLK can be used as the clock for counter
  - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software
- FULL interrupt and HALF interrupt can be generated for each channel using the compare data registers
  - Timer 0-7 can be used as PWM (Set the initial signal level)
  - Timer 0,3-7 can be used as a counter to count external signal (like trackball)
  - Timer 5 has separated interrupt
  - Timer 0-4 and timer 6-7 has one interrupt in common
  - OST uses interrupt 0, Timer 5 uses interrupt 1, and Timer 0-4/ 6-7 uses interrupt 2
- The difference between TCU1 and TCU2
  - TCU1: It cannot work in sleep mode, but operated easily
  - TCU2: It can work in sleep mode, but operated more complicated than TCU1

### 19.2 Pin Description

**Table 19-1 PWM Pins Description**

| Name      | I/O    | Description                 |
|-----------|--------|-----------------------------|
| PWM [7:0] | Output | PWM channel output signals. |

### 19.3 Register Description

In this section, we will describe the registers in timer. Following table lists all the registers definition. All timer register's 32bit address is physical address. And detailed function of each register will be described below.

| Name  | Description                         | RW | Reset Value | Address    | Access Size |
|-------|-------------------------------------|----|-------------|------------|-------------|
| TSTR  | Timer Status Register               | R  | 0x00000000  | 0x100020F0 | 32          |
| TSTSR | Timer Status Set Register           | W  | 0x????????  | 0x100020F4 | 32          |
| TSTCR | Timer Status Clear Register         | W  | 0x????????  | 0x100020F8 | 32          |
| TSR   | Timer STOP Register                 | R  | 0x00000000  | 0x1000201C | 32          |
| TSSR  | Timer STOP Set Register             | W  | 0x00000000  | 0x1000202C | 32          |
| TSCR  | Timer STOP Clear Register           | W  | 0x0000      | 0x1000203C | 32          |
| TER   | Timer Counter Enable Register       | R  | 0x0000      | 0x10002010 | 16          |
| TESR  | Timer Counter Enable Set Register   | W  | 0x????      | 0x10002014 | 16          |
| TECR  | Timer Counter Enable Clear Register | W  | 0x????      | 0x10002018 | 16          |
| TFR   | Timer Flag Register                 | R  | 0x003F003F  | 0x10002020 | 32          |
| TFSR  | Timer Flag Set Register             | W  | 0x????????  | 0x10002024 | 32          |
| TFCR  | Timer Flag Clear Register           | W  | 0x????????  | 0x10002028 | 32          |
| TMR   | Timer Mask Register                 | R  | 0x00000000  | 0x10002030 | 32          |
| TMSR  | Timer Mask Set Register             | W  | 0x????????  | 0x10002034 | 32          |
| TMCR  | Timer Mask Clear Register           | W  | 0x????????  | 0x10002038 | 32          |
| TDFR0 | Timer Data FULL Register 0          | RW | 0x????      | 0x10002040 | 16          |
| TDHR0 | Timer Data HALF Register 0          | RW | 0x????      | 0x10002044 | 16          |
| TCNT0 | Timer Counter 0                     | RW | 0x????      | 0x10002048 | 16          |
| TCSR0 | Timer Control Register 0            | RW | 0x0000      | 0x1000204C | 16          |
| TDFR1 | Timer Data FULL Register 1          | RW | 0x????      | 0x10002050 | 16          |
| TDHR1 | Timer Data HALF Register 1          | RW | 0x????      | 0x10002054 | 16          |
| TCNT1 | Timer Counter 1                     | RW | 0x????      | 0x10002058 | 16          |
| TCSR1 | Timer Control Register 1            | RW | 0x0000      | 0x1000205C | 16          |
| TDFR2 | Timer Data FULL Register 2          | RW | 0x????      | 0x10002060 | 16          |
| TDHR2 | Timer Data HALF Register 2          | RW | 0x????      | 0x10002064 | 16          |
| TCNT2 | Timer Counter 2                     | RW | 0x????      | 0x10002068 | 16          |
| TCSR2 | Timer Control Register 2            | RW | 0x0000      | 0x1000206C | 16          |
| TDFR3 | Timer Data FULL Register 3          | RW | 0x????      | 0x10002070 | 16          |
| TDHR3 | Timer Data HALF Register 3          | RW | 0x????      | 0x10002074 | 16          |
| TCNT3 | Timer Counter 3                     | RW | 0x????      | 0x10002078 | 16          |
| TCSR3 | Timer Control Register 3            | RW | 0x0000      | 0x1000207C | 16          |
| TDFR4 | Timer Data FULL Register 4          | RW | 0x????      | 0x10002080 | 16          |
| TDHR4 | Timer Data HALF Register 4          | RW | 0x????      | 0x10002084 | 16          |
| TCNT4 | Timer Counter 4                     | RW | 0x????      | 0x10002088 | 16          |
| TCSR4 | Timer Control Register 4            | RW | 0x0000      | 0x1000208C | 16          |
| TDFR5 | Timer Data FULL Register 5          | RW | 0x????      | 0x10002090 | 16          |
| TDHR5 | Timer Data HALF Register 5          | RW | 0x????      | 0x10002094 | 16          |
| TCNT5 | Timer Counter 5                     | RW | 0x????      | 0x10002098 | 16          |

|          |                                  |    |            |            |    |
|----------|----------------------------------|----|------------|------------|----|
| TCSR5    | Timer Control Register 5         | RW | 0x0000     | 0x1000209C | 16 |
| TDFR6    | Timer Data FULL Register 6       | RW | 0x????     | 0x100020A0 | 16 |
| TDHR6    | Timer Data HALF Register 6       | RW | 0x????     | 0x100020A4 | 16 |
| TCNT6    | Timer Counter 6                  | RW | 0x????     | 0x100020A8 | 16 |
| TCSR6    | Timer Control Register 6         | RW | 0x0000     | 0x100020AC | 16 |
| TDFR7    | Timer Data FULL Register 7       | RW | 0x????     | 0x100020B0 | 16 |
| TDHR7    | Timer Data HALF Register 7       | RW | 0x????     | 0x100020B4 | 16 |
| TCNT7    | Timer Counter 7                  | RW | 0x????     | 0x100020B8 | 16 |
| TCSR7    | Timer Control Register 7         | RW | 0x0000     | 0x100020BC | 16 |
| TCUMOD0  | Timer control mode Register 0    | RW | 0x????     | 0x10002100 | 16 |
| TCUMOD3  | Timer control mode Register 3    | RW | 0x????     | 0x10002110 | 16 |
| TCUMOD4  | Timer control mode Register 4    | RW | 0x????     | 0x10002120 | 16 |
| TCUMOD5  | Timer control mode Register 5    | RW | 0x????     | 0x10002130 | 16 |
| TFWD0    | Timer fifo write data Register 0 | RW | 0x???????? | 0x10002104 | 32 |
| TFWD3    | Timer fifo write data Register 3 | RW | 0x???????? | 0x10002114 | 32 |
| TFWD4    | Timer fifo write data Register 4 | RW | 0x???????? | 0x10002124 | 32 |
| TFWD5    | Timer fifo write data Register 5 | RW | 0x???????? | 0x10002134 | 32 |
| TFIFOSR0 | Timer fifo state Register        | R  | 0x??       | 0x10002108 | 6  |
| TFIFOSR3 | Timer fifo state Register        | R  | 0x??       | 0x10002118 | 6  |
| TFIFOSR4 | Timer fifo state Register        | R  | 0x??       | 0x10002128 | 6  |
| TFIFOSR5 | Timer fifo state Register        | R  | 0x??       | 0x10002138 | 6  |

### 19.3.1 Timer Control Register (TCSR)

The TCSR is a 16-bit read/write register. It contains the control bits for each channel. It is initialized to 0x00 by any reset.

|     |                                                             |                                                                                                            |
|-----|-------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
|     | TCSR0, TCSR1, TCSR2,<br>TCSR3, TCSR4, TCSR5<br>TCSR6, TCSR7 | 0x1000204C, 0x1000205C, 0x1000206C,<br>0x1000207C, 0x1000208C, 0x1000209C,<br>0x100020AC, 0x100020BC       |
| Bit |                                                             | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0                                                                      |
|     |                                                             | Reserved<br>BYPASS<br>CLRZ<br>SD<br>INITL<br>PWM_EN<br>PWM_IN_EN<br>PRESCALE<br>EXT_EN<br>RTC_EN<br>PCK_EN |
| RST |                                                             | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0                                                                            |

| Bits  | Name     | Description                                                                                                     | RW |
|-------|----------|-----------------------------------------------------------------------------------------------------------------|----|
| 15:11 | Reserved | Writing has no effect, read as zero.                                                                            | R  |
| 11    | BYPASS   | PWM bypass mode.<br>1: If PCK_EN = 1, this channel output PIXCLK;<br>If RTC_EN = 1, this channel output RTCCLK; | RW |

|       |           | <p>If EXT_EN = 1, this channel output EXTAL;<br/>Only one of those XXX_EN is permit available during one time.</p> <p>0: This BYPASS function disable.</p> <p>*Only when you want to let this PWM channel output some special clock (PCLK, RTCLK and EXTAL clock), you can set this register to 1. Otherwise keep it to 0. When you want to use BYPASS function, not forget offer clock supplies of relate channel (relate to register TSR, TSSR, TSCR).(the current version can not bypass pclk,and when bypass a clk,the relative timer will be stopped to reduce power consumption.)</p> |                        |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |
|-------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|-------|-------|-------------|---|---|---|-----------------------|---|---|---|-----------------------|---|---|---|------------------------|----|
| 10    | CLRZ      | <p>Clear counter to 0. It is only used in TCU2 mode.</p> <p>Writing 1 to this bit will clear the counter to 0. When the counter is finished setting to 0, it will be cleared by hardware.</p> <p>Writing 0 to this bit will be ignored.</p>                                                                                                                                                                                                                                                                                                                                                 | RW                     |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |
| 9     | SD        | <p>Shut Down (SD) the PWM output. It is only used in TCU1 mode.</p> <p>0: Graceful shutdown<br/>1: Abrupt shutdown</p> <p>Graceful shutdown: The output level for PWM output will keep the level only after the comparison match of FULL.(when write register to stop the timer,the timer will continue run until full match)</p> <p>Abrupt shutdown: The output level for PWM output will keep the level as soon as software write to stop the counter.(the conter will stop as soon as softerware write to stop the counter).</p>                                                         | RW                     |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |
| 8     | INITL     | <p>Selects an initial output level for PWM output.</p> <p>0: Low<br/>1: High</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RW                     |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |
| 7     | PWM_EN    | <p>PWM output pin control bit.</p> <p>0: PWM pin output disable, and the PWM pin will be set to the initial level according to INITL<br/>1: PWM pin output enable</p>                                                                                                                                                                                                                                                                                                                                                                                                                       | RW                     |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |
| 6     | PWM_IN_EN | <p>PWM input mode enable.</p> <p>Set to 1 to enable this function.</p> <p>In this function, PWM pin need to set GPIO as function0 to receive external signal, EXT_EN, RTC_EN, PCK_EN need to set 0. And TCNT became a counter to count this signal's both edges. (This bit in TCSR1, 2 are Reserved).</p>                                                                                                                                                                                                                                                                                   | RW                     |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |
| 5:3   | PRESCALE  | <p>These bits select the TCNT count clock frequency. Don't change this field when the channel is running.</p> <table border="1" data-bbox="491 1816 1278 1982"> <thead> <tr> <th>Bit 2</th> <th>Bit 1</th> <th>Bit 0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Internal clock: CLK/1</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Internal clock: CLK/4</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Internal clock: CLK/16</td> </tr> </tbody> </table>                                                                   | Bit 2                  | Bit 1 | Bit 0 | Description | 0 | 0 | 0 | Internal clock: CLK/1 | 0 | 0 | 1 | Internal clock: CLK/4 | 0 | 1 | 0 | Internal clock: CLK/16 | RW |
| Bit 2 | Bit 1     | Bit 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Description            |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |
| 0     | 0         | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Internal clock: CLK/1  |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |
| 0     | 0         | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Internal clock: CLK/4  |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |
| 0     | 1         | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Internal clock: CLK/16 |       |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |    |

|   |        |                                                                    |   |   |                          |    |
|---|--------|--------------------------------------------------------------------|---|---|--------------------------|----|
|   |        | 0                                                                  | 1 | 1 | Internal clock: CLK/64   |    |
|   |        | 1                                                                  | 0 | 0 | Internal clock: CLK/256  |    |
|   |        | 1                                                                  | 0 | 1 | Internal clock: CLK/1024 |    |
|   |        | 110~111                                                            |   |   | Reserved                 |    |
| 2 | EXT_EN | Select EXTAL as the timer clock input.<br>0: Disable<br>1: Enable  |   |   |                          | RW |
| 1 | RTC_EN | Select RTCCLK as the timer clock input.<br>0: Disable<br>1: Enable |   |   |                          | RW |
| 0 | PCK_EN | Select PCLK as the timer clock input.<br>0: Disable<br>1: Enable   |   |   |                          | RW |

**NOTE:** The input clock of timer and the PCLK should keep to the rules as follows:

| Input clock of timer: IN_CLK                                  | Clock generated from the frequency divider (PRESCALE): DIV_CLK |
|---------------------------------------------------------------|----------------------------------------------------------------|
| PCK_EN == 0, RTC_EN == 1 and EXT_EN == 0<br>(IN_CLK = RTCCLK) | $f_{DIV\_CLK} < \frac{1}{2} f_{PCLK}$                          |
| PCK_EN == 0, RTC_EN == 0 and EXT_EN == 1<br>(IN_CLK = EXTAL)  | $f_{DIV\_CLK} < \frac{1}{2} f_{PCLK}$                          |
| PCK_EN == 1, RTC_EN == 0 and EXT_EN == 0<br>(IN_CLK = PCLK)   | ANY                                                            |

### 19.3.2 Timer Data FULL Register (TDFR)

The comparison data FULL registers TDFR is used to store the data to be compared with the content of the up-counter TCNT. This register can be directly read and written. (Default: indeterminate) But it is not suggested changing when counter is working in TCU2 mode.

|     |                                                              |                                                                                                      |
|-----|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
|     | TDFR0, TDFR1, TDFR2,<br>TDFR3, TDFR4, TDFR5,<br>TDFR6, TDFR7 | 0x10002040, 0x10002050, 0x10002060,<br>0x10002070, 0x10002080, 0x10002090,<br>0x100020A0, 0x100020B0 |
| Bit |                                                              | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0                                                                |
|     |                                                              | TDFR                                                                                                 |
| RST |                                                              | ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?                                                                      |

### 19.3.3 Timer Data HALF Register (TDHR)

The comparison data HALF registers TDHR is used to store the data to be compared with the content of the up-counter TCNT. This register can be directly read and written. (Default: indeterminate) But it is not suggested changing when counter is working in TCU2 mode.

|     |                                                              |                                                                                                      |
|-----|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
|     | TDHR0, TDHR1, TDHR2,<br>TDHR3, TDHR4, TDHR5,<br>TDHR6, TDHR7 | 0x10002044, 0x10002054, 0x10002064,<br>0x10002074, 0x10002084, 0x10002094,<br>0x100020A4, 0x100020B4 |
| Bit |                                                              | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0                                                                |
|     |                                                              | TDHR                                                                                                 |
| RST |                                                              | ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?                                                                      |

### 19.3.4 Timer Counter (TCNT)

TCNT is a 16-bit read/write register. The up-counter TCNT can be reset to 0 by software and counts up using the prescaler output clock. When TCNT count up to equal to TDFR, it will reset to 0 and continue to count up.

**TCU1:** The counter data can be read out at any time. The data can be written at any time. This makes it possible to change the interrupt and/or clock output cycles temporarily. (Default: indeterminate)

**TCU2:** The counter data can be read out at any time, but you should read TSTR.REALn to check whether the data is real data or not. The data can only be written before counter is started, and the counter clock is pclk. But it can be cleared to 0 by setting TCSR.CLRZ to 1, and if the counter is really cleared, TCSR.CLRZ will be set to 0 by hardware.

|     |                                                              |                                                                                                      |
|-----|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
|     | TCNT0, TCNT1, TCNT2,<br>TCNT3, TCNT4, TCNT5,<br>TCNT6, TCNT7 | 0x10002048, 0x10002058, 0x10002068,<br>0x10002078, 0x10002088, 0x10002098,<br>0x100020A8, 0x100020B8 |
| Bit |                                                              | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0                                                                |
|     |                                                              | TCNT                                                                                                 |
| RST |                                                              | ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?                                                                      |

### 19.3.5 Timer Counter Enable Register (TER)

The TER is a 16-bit read-only register. It contains the counter enable control bits for each channel. It is initialized to 0x0000 by any reset. It can only be set by register TESR and TECR. Since the timer enable control bits are located in the same addresses, two or more timers can be started at the same time.

|            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |                   |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
|------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------|-------------------|----|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|
| <b>TER</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       | <b>0x10002010</b> |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
| <b>Bit</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|            |  |  |  |  |  |  |  |  |  |  |  |  |  |  | OSTEN | Reserved          |    |    |    |    |    |   |   | TCEN 7 | TCEN 6 | TCEN 5 | TCEN 4 | TCEN 3 | TCEN 2 | TCEN 1 | TCEN 0 |
| <b>RST</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       | 0                 | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

| Bits | Name     | Description                                                                   | RW |
|------|----------|-------------------------------------------------------------------------------|----|
| 15   | OSTEN    | Enable the counter in OST.<br>0: Stop counting up<br>1: Begin counting up     |    |
| 14:8 | Reserved | Writing has no effect, read as zero.                                          | R  |
| 7    | TCEN 7   | Enable the counter in timer 7.<br>0: Stop counting up<br>1: Begin counting up | R  |
| 6    | TCEN 6   | Enable the counter in timer 6.<br>0: Stop counting up<br>1: Begin counting up | R  |
| 5    | TCEN 5   | Enable the counter in timer 5.<br>0: Stop counting up<br>1: Begin counting up | R  |
| 4    | TCEN 4   | Enable the counter in timer 4.<br>0: Stop counting up<br>1: Begin counting up | R  |
| 3    | TCEN 3   | Enable the counter in timer 3.<br>0: Stop counting up<br>1: Begin counting up | R  |
| 2    | TCEN 2   | Enable the counter in timer 2.<br>0: Stop counting up<br>1: Begin counting up | R  |
| 1    | TCEN 1   | Enable the counter in timer 1.<br>1: Begin counting up<br>0: Stop counting up | R  |
| 0    | TCEN 0   | Enable the counter in timer 0.<br>0: Stop counting up<br>1: Begin counting up | R  |

### 19.3.6 Timer Counter Enable Set Register (TESR)

The TCCSR is a 32-bit write-only register. It contains the counter enable set bits for each channel. Since the timer enable control set bits are located in the same addresses, two or more timers can be started at the same time.



|      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |            |          |    |    |    |    |   |   |        |        |        |        |        |        |        |        |   |
|------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------|----------|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|---|
| TESR |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x10002014 |          |    |    |    |    |   |   |        |        |        |        |        |        |        |        |   |
| Bit  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15         | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |   |
|      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | OSTST      | Reserved |    |    |    |    |   |   | TCST 7 | TCST 6 | TCST 5 | TCST 4 | TCST 3 | TCST 2 | TCST 1 | TCST 0 |   |
| RST  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ?          | ?        | ?  | ?  | ?  | ?  | ? | ? | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ? |

| Bits | Name     | Description                                                   | RW |
|------|----------|---------------------------------------------------------------|----|
| 15   | OSTST    | Set OSTEN bit of TER.<br>0: Ignore<br>1: Set OSTEN bit to 1   | W  |
| 14:8 | Reserved | Writing has no effect, read as zero.                          | R  |
| 7    | TCST 7   | Set TCEN 7 bit of TER.<br>0: Ignore<br>1: Set TCEN 5 bit to 1 | W  |
| 6    | TCST 6   | Set TCEN 6 bit of TER.<br>0: Ignore<br>1: Set TCEN 5 bit to 1 | W  |
| 5    | TCST 5   | Set TCEN 5 bit of TER.<br>0: Ignore<br>1: Set TCEN 5 bit to 1 | W  |
| 4    | TCST 4   | Set TCEN 4 bit of TER.<br>1: Set TCEN 4 bit to 1<br>0: Ignore | W  |
| 3    | TCST 3   | Set TCEN 3 bit of TER.<br>0: Ignore<br>1: Set TCEN 3 bit to 1 | W  |
| 2    | TCST 2   | Set TCEN 2 bit of TER.<br>1: Set TCEN 2 bit to 1<br>0: Ignore | W  |
| 1    | TCST 1   | Set TCEN 1 bit of TER.<br>0: Ignore<br>1: Set TCEN 1 bit to 1 | W  |
| 0    | TCST 0   | Set TCEN 0 bit of TER.<br>0: Ignore<br>1: Set TCEN 0 bit to 1 | W  |

### 19.3.7 Timer Counter Enable Clear Register (TECR)

The TECR is a 32-bit write-only register. It contains the counter enable clear bits for each channel. Since the timer enable clear bits are located in the same addresses, two or more timers can be stop at the same time.

|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                   |          |    |    |    |    |   |   |        |        |        |        |        |        |        |        |   |
|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------------|----------|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|---|
| <b>TECR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x10002018</b> |          |    |    |    |    |   |   |        |        |        |        |        |        |        |        |   |
| <b>Bit</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15                | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |   |
|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  | OSTCL             | Reserved |    |    |    |    |   |   | TCCL 7 | TCCL 6 | TCCL 5 | TCCL 4 | TCCL 3 | TCCL 2 | TCCL 1 | TCCL 0 |   |
| <b>RST</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ?                 | ?        | ?  | ?  | ?  | ?  | ? | ? | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ? |

| Bits | Name     | Description                                                   | RW |
|------|----------|---------------------------------------------------------------|----|
| 15   | OSTCL    | Set OSTEN bit of TER.<br>0: Ignore<br>1: Set OSTEN 5 bit to 0 | W  |
| 14:8 | Reserved | Writing has no effect, read as zero.                          | R  |
| 7    | TCCL 7   | Set TCEN 7 bit of TER.<br>0: Ignore<br>1: Set TCEN 6 bit to 0 | W  |
| 6    | TCCL 6   | Set TCEN 7 bit of TER.<br>0: Ignore<br>1: Set TCEN 6 bit to 0 | W  |
| 5    | TCCL 5   | Set TCEN 5 bit of TER.<br>0: Ignore<br>1: Set TCEN 5 bit to 0 | W  |
| 4    | TCCL 4   | Set TCEN 4 bit of TER.<br>1: Set TCEN 4 bit to 0<br>0: Ignore | W  |
| 3    | TCCL 3   | Set TCEN 3 bit of TER.<br>0: Ignore<br>1: Set TCEN 3 bit to 0 | W  |
| 2    | TCCL 2   | Set TCEN 2 bit of TER.<br>1: Set TCEN 2 bit to 0<br>0: Ignore | W  |
| 1    | TCCL 1   | Set TCEN 1 bit of TER.<br>0: Ignore<br>1: Set TCEN 1 bit to 0 | W  |
| 0    | TCCL 0   | Set TCEN 0 bit of TER.<br>0: Ignore<br>1: Set TCEN 0 bit to 0 | W  |

### 19.3.8 Timer Flag Register (TFR)

The TFR is a 32-bit read-only register. It contains the comparison match flag bits for all the channels. It can also be set by register TFSR and TFCR. It is initialized to 0x00000000 by any reset.

| TFR |          |    |    |    |           |           |           |           |         |         |         |         |         |         |         | 0x10002020 |         |          |    |    |    |           |           |           |           |         |         |         |         |         |         |         |         |
|-----|----------|----|----|----|-----------|-----------|-----------|-----------|---------|---------|---------|---------|---------|---------|---------|------------|---------|----------|----|----|----|-----------|-----------|-----------|-----------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 31       | 30 | 29 | 28 | 27        | 26        | 25        | 24        | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16         | 15      | 14       | 13 | 12 | 11 | 10        | 9         | 8         | 7         | 6       | 5       | 4       | 3       | 2       | 1       | 0       |         |
|     | Reserved |    |    |    | FIFOFLAG5 | FIFOFLAG4 | FIFOFLAG3 | FIFOFLAG0 | HFLAG 7 | HFLAG 6 | HFLAG 5 | HFLAG 4 | HFLAG 3 | HFLAG 2 | HFLAG 1 | HFLAG 0    | OSTFLAG | Reserved |    |    |    | FIFOFLAG5 | FIFOFLAG4 | FIFOFLAG3 | FIFOFLAG0 | FFLAG 7 | FFLAG 6 | FFLAG 5 | FFLAG 4 | FFLAG 3 | FFLAG 2 | FFLAG 1 | FFLAG 0 |
| RST | 0        | 0  | 0  | 0  | 0         | 0         | 0         | 0         | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0          | 0       | 0        | 0  | 0  | 0  | 0         | 0         | 0         | 0         | 0       | 0       | 0       | 0       | 0       | 0       | 0       |         |

| Bits  | Name            | Description                                                                                   | RW |
|-------|-----------------|-----------------------------------------------------------------------------------------------|----|
| 31:28 | Reserved        | Writing has no effect, read as zero.                                                          | R  |
| 27:24 | FIFO empty FLAG | fifo empty flag:<br>0:fifo not empty<br>1:fifo empty                                          | R  |
| 23:16 | HFLAG 7~0       | HALF comparison match flag. (TCNT = TDHR)<br>0: Comparison not match<br>1: Comparison match   | R  |
| 15    | OSTFLAG         | OST comparison match flag. (OSTCNT = OSTDR)<br>0: Comparison not match<br>1: Comparison match | R  |
| 14:12 | Reserved        | Writing has no effect, read as zero.                                                          | R  |
| 11:8  | FIFOFLAG 3~0    | FIFO comparison match flag. (TCNT = TFWD)<br>0: Comparison not match<br>1: Comparison match   | R  |
| 7:0   | FFLAG 7~0       | FULL comparison match flag. (TCNT = TDFR)<br>0: Comparison not match<br>1: Comparison match   | R  |

### 19.3.9 Timer Flag Set Register (TFSR)

The TFSR is a 32-bit write-only register. It contains the comparison match flag set bits for all the channels.

| TFSR |          |    |    |    |          |          |          |          |        |        |        |        |        |          |        | 0x10002024 |          |    |    |    |          |          |          |          |        |        |        |        |        |          |        |   |
|------|----------|----|----|----|----------|----------|----------|----------|--------|--------|--------|--------|--------|----------|--------|------------|----------|----|----|----|----------|----------|----------|----------|--------|--------|--------|--------|--------|----------|--------|---|
| Bit  | 31       | 30 | 29 | 28 | 27       | 26       | 25       | 24       | 23     | 22     | 21     | 20     | 19     | 18       | 17     | 16         | 15       | 14 | 13 | 12 | 11       | 10       | 9        | 8        | 7      | 6      | 5      | 4      | 3      | 2        | 1      | 0 |
|      | Reserved |    |    |    | FIFOEST5 | FIFOEST4 | FIFOEST3 | FIFOEST0 | HFST 7 | HFST 6 | HFST 5 | HFST 4 | HFST 3 | Reserved | HFST 0 | OSTFST     | Reserved |    |    |    | FIFOEST5 | FIFOEST4 | FIFOEST3 | FIFOEST0 | FFST 7 | FFST 6 | FFST 5 | FFST 4 | FFST 3 | Reserved | FFST 0 |   |
| RST  | ?        | ?  | ?  | ?  | ?        | ?        | ?        | ?        | ?      | ?      | ?      | ?      | ?      | ?        | ?      | ?          | ?        | ?  | ?  | ?  | ?        | ?        | ?        | ?        | ?      | ?      | ?      | ?      | ?      | ?        | ?      | ? |

| Bits  | Name           | Description                                                          | RW |
|-------|----------------|----------------------------------------------------------------------|----|
| 31:28 | Reserved       | Writing has no effect, read as zero.                                 | R  |
| 27:24 | FIFOEST<br>3~0 | Set FIFOEFLAG bit of TFR<br>0: Ignore; 1: Set FIFOEFLAG n bit to 1.  | W  |
| 23:16 | HFST 7~0       | Set HFLAG n bit of TFR.<br>0: Ignore<br>1: Set HFLAG n bit to 1      | W  |
| 15    | OSTFST         | Set OSTFLAG n bit of TFR.<br>0: Ignore<br>1: Set OSTFLAG n bit to 1  | W  |
| 14:12 | Reserved       | Writing has no effect, read as zero.                                 | R  |
| 11:8  | FIFOST 3~0     | Set FIFOFLAG n bit of TFR.<br>0: Ignore; 1: Set FIFOFLAG n bit to 1. | W  |
| 7:0   | FFST 7~0       | Set FFLAG n bit of TFR.<br>0: Ignore<br>1: Set FFLAG n bit to 1      | W  |

### 19.3.10 Timer Flag Clear Register (TFCR)

The TFCR is a 32-bit write-only register. It contains the comparison match flag clear bits for all the channels.

| TFCR |          |    |    |    |          |          |          |          |       |       |       |       |       |       |       | 0x10002028 |        |          |    |    |    |         |         |         |         |       |       |       |       |       |       |       |       |
|------|----------|----|----|----|----------|----------|----------|----------|-------|-------|-------|-------|-------|-------|-------|------------|--------|----------|----|----|----|---------|---------|---------|---------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit  | 31       | 30 | 29 | 28 | 27       | 26       | 25       | 24       | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16         | 15     | 14       | 13 | 12 | 11 | 10      | 9       | 8       | 7       | 6     | 5     | 4     | 3     | 2     | 1     | 0     |       |
|      | Reserved |    |    |    | FIFOECL5 | FIFOECL4 | FIFOECL3 | FIFOECL0 | HFCL7 | HFCL6 | HFCL5 | HFCL4 | HFCL3 | HFCL2 | HFCL1 | HFCL0      | OSTFCL | Reserved |    |    |    | FIFOCL5 | FIFOCL4 | FIFOCL3 | FIFOCL0 | FFCL7 | FFCL6 | FFCL5 | FFCL4 | FFCL3 | FFCL2 | FFCL1 | FFCL0 |
| RST  | ?        | ?  | ?  | ?  | ?        | ?        | ?        | ?        | ?     | ?     | ?     | ?     | ?     | ?     | ?     | ?          | ?      | ?        | ?  | ?  | ?  | ?       | ?       | ?       | ?       | ?     | ?     | ?     | ?     | ?     | ?     | ?     |       |

| Bits  | Name           | Description                                                             | RW |
|-------|----------------|-------------------------------------------------------------------------|----|
| 31:28 | Reserved       | Writing has no effect, read as zero.                                    | R  |
| 27:24 | FIFOECL<br>3~0 | Set FIFOEFLAG n bit of TFR.<br>0: Ignore<br>1: Set FIFOEFLAG n bit to 0 |    |
| 23:16 | HFCL 7~0       | Set HFLAG n bit of TFR.<br>0: Ignore<br>1: Set FFLAG n bit to 0         | W  |
| 15    | OSTFCL         | Set OSTFLAG n bit of TFR.<br>0: Ignore<br>1: Set OSTFLAG n bit to 0     | W  |
| 14:12 | Reserved       | Writing has no effect, read as zero.                                    | R  |

|      |            |                                                                       |   |
|------|------------|-----------------------------------------------------------------------|---|
| 11:8 | FIFOCL 3~0 | Set FIFOFLAG n bit of TFR.<br>0: Ignore<br>1: Set FIFOFLAG n bit to 0 | W |
| 7:0  | FFCL 7~0   | Set FFLAG n bit of TFR.<br>0: Ignore<br>1: Set FFLAG n bit to 0       | W |

### 19.3.11 Timer Mast Register (TMR)

The TMR is a 32-bit read-only register. It contains the comparison match flag bits for all the channels. It is initialized to 0x003F003F by any reset. It can only be set by register TMSR and TMCR.

| TMR |          |    |    |    |            |            |            |            |         |         |         |         |         |         |         | 0x10002030 |         |          |    |    |    |           |           |           |           |         |         |         |         |         |         |         |         |
|-----|----------|----|----|----|------------|------------|------------|------------|---------|---------|---------|---------|---------|---------|---------|------------|---------|----------|----|----|----|-----------|-----------|-----------|-----------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit | 31       | 30 | 29 | 28 | 27         | 26         | 25         | 24         | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16         | 15      | 14       | 13 | 12 | 11 | 10        | 9         | 8         | 7         | 6       | 5       | 4       | 3       | 2       | 1       | 0       |         |
|     | Reserved |    |    |    | FIFOEMASK5 | FIFOEMASK4 | FIFOEMASK3 | FIFOEMASK0 | HMASK 7 | HMASK 6 | HMASK 5 | HMASK 4 | HMASK 3 | HMASK 2 | HMASK 1 | HMASK 0    | OSTMASK | Reserved |    |    |    | FIFOMASK5 | FIFOMASK4 | FIFOMASK3 | FIFOMASK0 | FMASK 7 | FMASK 6 | FMASK 5 | FMASK 4 | FMASK 3 | FMASK 2 | FMASK 1 | FMASK 0 |
| RST | 0        | 0  | 0  | 0  | 0          | 0          | 0          | 0          | 0       | 0       | 1       | 1       | 1       | 1       | 1       | 1          | 0       | 0        | 0  | 0  | 0  | 0         | 0         | 0         | 0         | 0       | 1       | 1       | 1       | 1       | 1       | 1       |         |

| Bits  | Name          | Description                                                                                                           | RW |
|-------|---------------|-----------------------------------------------------------------------------------------------------------------------|----|
| 31:28 | Reserved      | Writing has no effect, read as zero.                                                                                  | R  |
| 27:24 | FIFOEMASK 3~0 | FIFO empty interrupt mask.<br>0:empty interrupt not mask<br>1:empty interrupt mask                                    |    |
| 23:16 | HMASK 7~0     | HALF comparison match interrupt mask.<br>0: Comparison match interrupt not mask<br>1: Comparison match interrupt mask | R  |
| 15    | OSTMASK       | OST comparison match interrupt mask.<br>0: Comparison match interrupt not mask<br>1: Comparison match interrupt mask  | R  |
| 14:12 | Reserved      | Writing has no effect, read as zero.                                                                                  | R  |
| 11:8  | FIFOMASK 3~0  | FIFO comparison match interrupt mask.<br>0: Comparison match interrupt not mask<br>1: Comparison match interrupt mask | R  |
| 7:0   | FMASK 7~0     | FULL comparison match interrupt mask.<br>0: Comparison match interrupt not mask<br>1: Comparison match interrupt mask | R  |

### 19.3.12 Timer Mask Set Register (TMSR)

The TMSR is a 32-bit write-only register. It contains the comparison match flag set bits for all the channels.

| TMSR |          |    |    |    |           |           |           |           |        |        |        |        |        |        |        | 0x10002034 |        |          |    |    |    |          |          |          |          |        |        |        |        |        |        |        |        |
|------|----------|----|----|----|-----------|-----------|-----------|-----------|--------|--------|--------|--------|--------|--------|--------|------------|--------|----------|----|----|----|----------|----------|----------|----------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit  | 31       | 30 | 29 | 28 | 27        | 26        | 25        | 24        | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16         | 15     | 14       | 13 | 12 | 11 | 10       | 9        | 8        | 7        | 6      | 5      | 4      | 3      | 2      | 1      | 0      |        |
|      | Reserved |    |    |    | FIFOEMST5 | FIFOEMST4 | FIFOEMST3 | FIFOEMST0 | HMST 7 | HMST 6 | HMST 5 | HMST 4 | HMST 3 | HMST 2 | HMST 1 | HMST 0     | OSTMST | Reserved |    |    |    | FIFOMST5 | FIFOMST4 | FIFOMST3 | FIFOMST0 | FMST 7 | FMST 6 | FMST 5 | FMST 4 | FMST 3 | FMST 2 | FMST 1 | FMST 0 |
| RST  | ?        | ?  | ?  | ?  | ?         | ?         | ?         | ?         | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?          | ?      | ?        | ?  | ?  | ?  | ?        | ?        | ?        | ?        | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      |

| Bits  | Name            | Description                                                            | RW |
|-------|-----------------|------------------------------------------------------------------------|----|
| 31:28 | Reserved        | Writing has no effect, read as zero.                                   | R  |
| 27:24 | FIFOEMST<br>3~0 | Set FIFOEMST n bit of TMR.<br>0: Ignore<br>1: Set FIFOEMASK n bit to 1 |    |
| 23:16 | HMST 7~0        | Set HMASK n bit of TMR.<br>0: Ignore<br>1: Set HMASK n bit to 1        | W  |
| 15    | OSTMST          | Set OSTMASK n bit of TMR.<br>0: Ignore<br>1: Set OSTMASK n bit to 1    | W  |
| 14:12 | Reserved        | Writing has no effect, read as zero.                                   | R  |
| 11:8  | FIFOMST<br>3~0  | Set FIFOMST n bit of TMR.<br>0: Ignore<br>1: Set FMASK n bit to 1      | W  |
| 7:0   | FMST 7~0        | Set FMASK n bit of TMR.<br>0: Ignore<br>1: Set FMASK n bit to 1        | W  |

### 19.3.13 Timer Mask Clear Register (TMCR)

The TMCR is a 32-bit write-only register. It contains the comparison match flag clear bits for all the channels.

| TMCR |          |    |    |    |           |           |           |           |        |        |        |        |        |        |        | 0x10002038 |        |          |    |    |    |          |          |          |          |        |        |        |        |        |        |        |        |
|------|----------|----|----|----|-----------|-----------|-----------|-----------|--------|--------|--------|--------|--------|--------|--------|------------|--------|----------|----|----|----|----------|----------|----------|----------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit  | 31       | 30 | 29 | 28 | 27        | 26        | 25        | 24        | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16         | 15     | 14       | 13 | 12 | 11 | 10       | 9        | 8        | 7        | 6      | 5      | 4      | 3      | 2      | 1      | 0      |        |
|      | Reserved |    |    |    | FIFOEMCL5 | FIFOEMCL4 | FIFOEMCL3 | FIFOEMCL0 | HMCL 7 | HMCL 6 | HMCL 5 | HMCL 4 | HMCL 3 | HMCL 2 | HMCL 1 | HMCL 0     | OSTMCL | Reserved |    |    |    | FIFOMCL5 | FIFOMCL4 | FIFOMCL3 | FIFOMCL0 | FMCL 7 | FMCL 6 | FMCL 5 | FMCL 4 | FMCL 3 | FMCL 2 | FMCL 1 | FMCL 0 |
| RST  | ?        | ?  | ?  | ?  | ?         | ?         | ?         | ?         | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?          | ?      | ?        | ?  | ?  | ?  | ?        | ?        | ?        | ?        | ?      | ?      | ?      | ?      | ?      | ?      | ?      |        |

| Bits  | Name            | Description                                                            | RW |
|-------|-----------------|------------------------------------------------------------------------|----|
| 31:28 | Reserved        | Writing has no effect, read as zero.                                   | R  |
| 27:24 | FIFOEMCL<br>3~0 | Set FIFOEMST n bit of TMR.<br>0: Ignore<br>1: Set FIFOEMASK n bit to 0 |    |
| 23:16 | HMCL 7~0        | Set HMASK n bit of TMR.<br>0: Ignore<br>1: Set HMASK n bit to 0        | W  |
| 15    | OSTMCL          | Set OSTMASK n bit of TMR.<br>0: Ignore<br>1: Set OSTMASK n bit to 0    | W  |
| 14:12 | Reserved        | Writing has no effect, read as zero.                                   | R  |
| 11:8  | FIFOMCL<br>3~0  | Set FIFOMCL n bit of TMR.<br>0: Ignore<br>1: Set FIFOMCL n bit to 0    | W  |
| 7:0   | FMCL 7~0        | Set FMASK n bit of TMR.<br>0: Ignore<br>1: Set FMASK n bit to 0        | W  |

### 19.3.14 Timer Stop Register (TSR)

The TSR is a 32-bit read-only register. It contains the timer stop control bits for each channel, WDT and OST. It is initialized to 0x00000000 by any reset. It can only be set by register TSSR and TSCR. If set, clock supplies to timer n / WDT / OST is stopped, and registers of the timer / WDT / OST cannot be accessed also.

| TSR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |      | 0x1000201C |          |    |    |    |    |    |   |   |   |   |       |       |       |       |       |       |       |       |
|-----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------------|----------|----|----|----|----|----|---|---|---|---|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17   | 16         | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5     | 4     | 3     | 2     | 1     | 0     |       |       |
|     | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | WDTS | OSTS       | Reserved |    |    |    |    |    |   |   |   |   | STOP7 | STOP6 | STOP5 | STOP4 | STOP3 | STOP2 | STOP1 | STOP0 |
| RST | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0          | 0        | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0     | 0     | 0     | 0     | 0     | 0     | 0     |       |

| Bits  | Name     | Description                                                                         | RW |
|-------|----------|-------------------------------------------------------------------------------------|----|
| 31:17 | Reserved | Writing has no effect, read as zero.                                                | R  |
| 16    | WDTS     | 0: The clock supplies to WDT is supplied<br>1: The clock supplies to WDT is stopped | R  |
| 15    | OSTS     | 0: The clock supplies to OST is supplied<br>1: The clock supplies to OST is stopped | R  |
| 14:8  | Reserved | Writing has no effect, read as zero.                                                | R  |

|   |        |                                                                                             |   |
|---|--------|---------------------------------------------------------------------------------------------|---|
| 7 | STOP 7 | 0: The clock supplies to timer 7 is supplied<br>1: The clock supplies to timer 7 is stopped | R |
| 6 | STOP 6 | 0: The clock supplies to timer 6 is supplied<br>1: The clock supplies to timer 6 is stopped | R |
| 5 | STOP 5 | 0: The clock supplies to timer 5 is supplied<br>1: The clock supplies to timer 5 is stopped | R |
| 4 | STOP 4 | 0: The clock supplies to timer 4 is supplied<br>1: The clock supplies to timer 4 is stopped | R |
| 3 | STOP 3 | 0: The clock supplies to timer 3 is supplied<br>1: The clock supplies to timer 3 is stopped | R |
| 2 | STOP 2 | 0: The clock supplies to timer 2 is supplied<br>1: The clock supplies to timer 2 is stopped | R |
| 1 | STOP 1 | 0: The clock supplies to timer 1 is supplied<br>1: The clock supplies to timer 1 is stopped | R |
| 0 | STOP 0 | 0: The clock supplies to timer 0 is supplied<br>1: The clock supplies to timer 0 is stopped | R |

### 19.3.15 Timer Stop Set Register (TSSR)

The TCSR is an 32-bit write-only register. It contains the timer stop set bits for each channel, WDT and OST. Since the timer stop control set bits are located in the same addresses, two or more timers can be started at the same time.

|             |          |    |    |    |    |    |    |    |    |    |    |    |    |    |       |                   |          |    |    |    |    |    |   |   |   |   |   |   |   |   |       |       |       |       |       |       |       |       |
|-------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------------------|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>TSSR</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |    |       | <b>0x1000202C</b> |          |    |    |    |    |    |   |   |   |   |   |   |   |   |       |       |       |       |       |       |       |       |
| <b>Bit</b>  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17    | 16                | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |       |       |       |       |       |       |
|             | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | WDTSS | OSTSS             | Reserved |    |    |    |    |    |   |   |   |   |   |   |   |   | STPS7 | STPS6 | STPS5 | STPS4 | STPS3 | STPS2 | STPS1 | STPS0 |
| <b>RST</b>  | ?        | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?     | ?                 | ?        | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ?     | ?     | ?     | ?     |       |       |       |       |

| Bits  | Name     | Description                                                   | RW |
|-------|----------|---------------------------------------------------------------|----|
| 31:17 | Reserved | Writing has no effect, read as zero.                          | R  |
| 16    | WDTSS    | Set WDTS bit of TSR.<br>0: Ignore<br>1: Set WDTS bit to 1     | W  |
| 15    | OSTSS    | Set OSTS bit of TSR.<br>0: Ignore<br>1: Set OSTS bit to 1     | W  |
| 14:8  | Reserved | Writing has no effect, read as zero.                          | R  |
| 7     | STPS 7   | Set STOP 7 bit of TSR.<br>0: Ignore<br>1: Set STOP 7 bit to 1 | W  |



|   |        |                                                               |   |
|---|--------|---------------------------------------------------------------|---|
| 6 | STPS 6 | Set STOP 6 bit of TSR.<br>0: Ignore<br>1: Set STOP 6 bit to 1 | W |
| 5 | STPS 5 | Set STOP 5 bit of TSR.<br>0: Ignore<br>1: Set STOP 5 bit to 1 | W |
| 4 | STPS 4 | Set STOP 4 bit of TSR.<br>1: Set STOP 4 bit to 1<br>0: Ignore | W |
| 3 | STPS 3 | Set STOP 3 bit of TSR.<br>0: Ignore<br>1: Set STOP 3 bit to 1 | W |
| 2 | STPS 2 | Set STOP 2 bit of TSR.<br>0: Ignore<br>1: Set STOP 2 bit to 1 | W |
| 1 | STPS 1 | Set STOP 1 bit of SR.<br>0: Ignore<br>1: Set STOP 1 bit to 1  | W |
| 0 | STPS 0 | Set STOP 0 bit of TSR.<br>0: Ignore<br>1: Set STOP 0 bit to 1 | W |

### 19.3.16 Timer Stop Clear Register (TSCR)

The TSCR is an 32-bit write-only register. It contains the timer stop clear bits for each channel, WDT and OST. Since the timer stop clear bits are located in the same addresses, two or more timers can be stop at the same time.

| TSCR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |       | 0x1000203C |          |    |    |    |    |    |   |   |   |   |   |   |   |   |       |       |       |       |       |       |       |       |
|------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|------------|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17    | 16         | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |       |       |       |       |       |       |
|      | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | WDTSC | OSTSC      | Reserved |    |    |    |    |    |   |   |   |   |   |   |   |   | STPC7 | STPC6 | STPC5 | STPC4 | STPC3 | STPC2 | STPC1 | STPC0 |
| RST  | ?        | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?     | ?          | ?        | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ?     | ?     |       |       |       |       |       |       |

| Bits  | Name     | Description                                               | RW |
|-------|----------|-----------------------------------------------------------|----|
| 31:17 | Reserved | Writing has no effect, read as zero.                      | R  |
| 16    | WDTSC    | Set WDTS bit of TSR.<br>0: Ignore<br>1: Set WDTS bit to 0 | W  |
| 15    | OSTSC    | Set OSTS bit of TSR.<br>0: Ignore                         | W  |

|      |          |                                                               |   |
|------|----------|---------------------------------------------------------------|---|
|      |          | 1: Set OSTS bit to 0                                          |   |
| 14:8 | Reserved | Writing has no effect, read as zero.                          | R |
| 7    | STPC 7   | Set STOP 7 bit of TSR.<br>0: Ignore<br>1: Set STOP 7 bit to 0 | W |
| 6    | STPC 6   | Set STOP 6 bit of TSR.<br>0: Ignore<br>1: Set STOP 6 bit to 0 | W |
| 5    | STPC 5   | Set STOP 5 bit of TSR.<br>0: Ignore<br>1: Set STOP 5 bit to 0 | W |
| 4    | STPC 4   | Set STOP 4 bit of TSR.<br>0: Ignore<br>1: Set STOP 4 bit to 0 | W |
| 3    | STPC 3   | Set STOP 3 bit of TSR.<br>0: Ignore<br>1: Set STOP 3 bit to 0 | W |
| 2    | STPC 2   | Set STOP 2 bit of TSR.<br>0: Ignore<br>1: Set STOP 2 bit to 0 | W |
| 1    | STPC 1   | Set STOP 1 bit of TSR.<br>0: Ignore<br>1: Set STOP 1 bit to 0 | W |
| 0    | STPC 0   | Set STOP 0 bit of TSR.<br>0: Ignore<br>1: Set STOP 0 bit to 0 | W |

### 19.3.17 Timer Status Register (TSTR)

The TSTR is a 32-bit read-only register. It contains the status of channel in TCU2 mode. The register can be written by setting register TSTSR and TSTCR.

|            |            |    |    |    |    |    |    |    |    |    |    |    |    |    |       |       |                   |    |    |    |    |    |   |   |   |   |   |   |   |   |       |       |          |
|------------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|-------|-------|----------|
|            | <b>TSR</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |       |       | <b>0x100020F0</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |       |       |          |
| <b>Bit</b> | 31         | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17    | 16    | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |          |
|            | Reserved   |    |    |    |    |    |    |    |    |    |    |    |    |    | REAL2 | REAL1 | Reserved          |    |    |    |    |    |   |   |   |   |   |   |   |   | BUSY2 | BUSY1 | Reserved |
| <b>RST</b> | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0                 | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0     | 0     | 0        |

| Bits  | Name     | Description                                       | RW |
|-------|----------|---------------------------------------------------|----|
| 31:19 | Reserved | Writing has no effect, read as zero.              | R  |
| 18    | REAL 2   | 0: The value read from counter 2 is a false value | R  |

|      |          |                                                                                                       |   |
|------|----------|-------------------------------------------------------------------------------------------------------|---|
|      |          | 1: The value read from counter 2 is a real value                                                      |   |
| 17   | REAL1    | 0: The value read from counter 1 is a false value<br>1: The value read from counter 1 is a real value | R |
| 16:3 | Reserved | Writing has no effect, read as zero.                                                                  | R |
| 2    | BUSY 2   | 0: The counter 2 is ready now<br>1: The counter 2 is busy now                                         | R |
| 1    | BUSY1    | 0: The counter 1 is ready now<br>1: The counter 1 is busy now                                         | R |
| 0    | Reserved | Writing has no effect, read as zero.                                                                  | R |

### 19.3.18 Timer Status Set Register (TSTSR)

The TSTSR is a 32-bit write-only register. It contains the timer status set bits for each channel.

| TSTSR |          |    |    |    |    |    |    |    |    |    |    |    |    |         |         |          | 0x100020F4 |    |    |    |    |    |   |   |   |   |   |   |         |         |          |   |   |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|----------|------------|----|----|----|----|----|---|---|---|---|---|---|---------|---------|----------|---|---|
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17      | 16       | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3       | 2       | 1        | 0 |   |
|       | Reserved |    |    |    |    |    |    |    |    |    |    |    |    | REALS 2 | REALS 1 | Reserved |            |    |    |    |    |    |   |   |   |   |   |   | BUSYS 2 | BUSYS 1 | Reserved |   |   |
| RST   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0       | 0        | 0          | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0       | 0       | 0        | 0 | 0 |

| Bits  | Name     | Description                                                    | RW |
|-------|----------|----------------------------------------------------------------|----|
| 31:19 | Reserved | Writing has no effect, read as zero.                           | R  |
| 18    | REALS 2  | Set REAL 2 bit of TSTR.<br>0: Ignore<br>1: Set REAL 2 bit to 1 | R  |
| 17    | REALS 1  | Set REAL 1 bit of TSTR.<br>0: Ignore<br>1: Set REAL 1 bit to 1 | R  |
| 16:3  | Reserved | Writing has no effect, read as zero.                           | R  |
| 2     | BUSYS 2  | Set BUSY 2 bit of TSTR.<br>0: Ignore<br>1: Set BUSY 2 bit to 1 | R  |
| 1     | BUSYS 1  | Set BUSY 1 bit of TSTR.<br>0: Ignore<br>1: Set BUSY 1 bit to 1 | R  |
| 0     | Reserved | Writing has no effect, read as zero.                           | R  |

### 19.3.19 Timer Status Clear Register (TSTCR)

The TSTCR is a 32-bit write-only register. It contains the timer status clear bits for each channel.

|              |          |    |    |    |    |    |    |    |    |    |    |    |    |         |         |          |                   |    |    |    |    |    |   |   |   |   |   |   |         |         |          |   |   |
|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|----------|-------------------|----|----|----|----|----|---|---|---|---|---|---|---------|---------|----------|---|---|
| <b>TSTCR</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |         |         |          | <b>0x100020F8</b> |    |    |    |    |    |   |   |   |   |   |   |         |         |          |   |   |
| <b>Bit</b>   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17      | 16       | 15                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3       | 2       | 1        | 0 |   |
|              | Reserved |    |    |    |    |    |    |    |    |    |    |    |    | REALC 2 | REALC 1 | Reserved |                   |    |    |    |    |    |   |   |   |   |   |   | BUSYC 2 | BUSYC 1 | Reserved |   |   |
| <b>RST</b>   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0       | 0        | 0                 | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0       | 0       | 0        | 0 | 0 |

| Bits  | Name     | Description                                                        | RW |
|-------|----------|--------------------------------------------------------------------|----|
| 31:19 | Reserved | Writing has no effect, read as zero.                               | R  |
| 18    | REALC 2  | Clear REAL 2 bit of TSTR.<br>0: Ignore<br>1: Clear REAL 2 bit to 1 | R  |
| 17    | REALC 1  | Clear REAL 1 bit of TSTR.<br>0: Ignore<br>1: Clear REAL 1 bit to 1 | R  |
| 16:3  | Reserved | Writing has no effect, read as zero.                               | R  |
| 2     | BUSYC 2  | Clear BUSY 2 bit of TSTR.<br>0: Ignore<br>1: Clear BUSY 2 bit to 1 | R  |
| 1     | BUSYC 1  | Clear BUSY 1 bit of TSTR.<br>0: Ignore<br>1: Clear BUSY 1 bit to 1 | R  |
| 0     | Reserved | Writing has no effect, read as zero.                               | R  |

### 19.3.20 Timer control mode Register (TCUMOD)

The TCUMOD is a 32-bit read write register. It contains the fifo control signal when TCU work in the TCU fifo mode , and only for the TCU0 、TCU3、TCU4、TCU5.

|                          |          |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |                                |          |    |    |             |          |          |         |   |   |   |   |   |   |   |   |   |
|--------------------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|--------------------------------|----------|----|----|-------------|----------|----------|---------|---|---|---|---|---|---|---|---|---|
| <b>TCUMOD0 ,TCUMOD3,</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    | <b>0x10002100, 0x10002110,</b> |          |    |    |             |          |          |         |   |   |   |   |   |   |   |   |   |
| <b>TCUMOD4,TCUMOD5</b>   |          |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    | <b>0x10002120, 0x10002130</b>  |          |    |    |             |          |          |         |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>               | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17 | 16 | 15                             | 14       | 13 | 12 | 11          | 10       | 9        | 8       | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|                          | Reserved |    |    |    |    |    |    |    |    |    |    |    |    | cyc_num |    |    | cyc_st                         | fifo_num |    |    | fifo_num_st | al_empty | fifo_clr | FIFOMOD |   |   |   |   |   |   |   |   |   |
| <b>RST</b>               | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0                              | 0        | 0  | 0  | 0           | 0        | 0        | 0       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                      | RW |
|-------|----------|------------------------------------------------------------------|----|
| 31:16 | Reserved | Writing has no effect, read as zero.                             | R  |
| 15:10 | cyc_num  | Set the fifo cycle numbers.<br>when the TCU work in FIFO mode 2. | RW |

|     |             |                                                                                           |    |
|-----|-------------|-------------------------------------------------------------------------------------------|----|
| 9   | cyc_st      | Select the FIFO mode.<br>0: work in FIFO mode 1<br>1: work in FIFO mode 2                 | RW |
| 8:4 | fifo_num    | Set the fifo data numbers can read out of the fifo.<br>when the TCU work in FIFO mode 2.  | RW |
| 3   | fifo_num_st | Select the FIFO mode.<br>0: work in FIFO mode 1<br>1: work in FIFO mode 2                 | RW |
| 2   | al_empty    | please keep this bit value is 0.                                                          | RW |
| 1   | fifo_clr    | 0: do nothing<br>1:clr the fifo                                                           |    |
| 0   | FIFOMOD     | Set the TCU work in fifo mode.<br>0: TCU work in nonfifo mode<br>1: TCU work in fifo mode | RW |

### 19.3.21 Timer fifo write data (TFWD)

The comparison data FIFO registers TFWD is used to store the data to be compared with the content of the up-counter TCNT. This register can be directly read and written(Default: 0),but only can read the 16 bit data at once. And Only use in the fifo mode.

|     |                                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                                   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-----|--------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|     | TFWD0 , TFWD3,<br>TFWD4, TFWD5 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10002104, 0x10002114,<br>0x10002124, 0x10002134 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit | 31                             | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                                                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|     | TFWD_HIGH                      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TFWD_LOW                                          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST | 0                              | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                                                 | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 19.3.22 Timer fifo state Register (TFIFOSR)

The TFIFOSR is a 32-bit write-only register. It contains the timer fifo status bits for TCU work in fifo mode.

|     |                                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                                   |    |    |    |    |    |            |   |   |   |             |   |   |   |   |   |
|-----|--------------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------------------------------------------|----|----|----|----|----|------------|---|---|---|-------------|---|---|---|---|---|
|     | TFIFOSR0 , TFIFOSR3,<br>TFIFOSR4, TFIFOSR5 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10002108, 0x10002118,<br>0x10002128, 0x10002138 |    |    |    |    |    |            |   |   |   |             |   |   |   |   |   |
| Bit | 31                                         | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                                                | 14 | 13 | 12 | 11 | 10 | 9          | 8 | 7 | 6 | 5           | 4 | 3 | 2 | 1 | 0 |
|     | reserve                                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | cyc_timer                                         |    |    |    |    |    | fifo_entry |   |   |   | fifo_w_flag |   |   |   |   |   |
| RST | 0                                          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                                                 | 0  | 0  | 0  | 0  | 0  | 0          | 0 | 0 | 0 | 0           | 0 | 0 | 0 | 0 | 0 |

| Bits | Name | Description | RW |
|------|------|-------------|----|
|------|------|-------------|----|

|       |                    |                                                         |   |
|-------|--------------------|---------------------------------------------------------|---|
| 31:12 | Reserved           | Writing has no effect, read as zero.                    | R |
| 11: 6 | <b>cyc_timer</b>   | record the number of cycs had been done.(valid in fifo) | R |
| 5 : 1 | <b>fifo_entry</b>  | record fifo's read address pointer                      | R |
| 0     | <b>fifo_w_flag</b> | record the fifo's state                                 | R |

## 19.4 Operation

### 19.4.1 Basic Operation in TCU1 Mode

The value of TDFR should be bigger than TDHR, and the minimum settings are TDHR = 0 and TDFR = 1. In this case, the timer output clock cycle is the input clock  $\times 1/2$ . If TDHR > TDFR, no comparison TFHR signal is generated.

Before the timer counter begin to count up, we need to do as follows:

If you want to use PWM you should set TCSR.PWM\_EN to be 0 before you initial TCU.

- 1 Initial the configuration.
  - a Writing TCSR.INITL to initialize PWM output level.
  - b Writing TCSR.SD to setting the shutdown mode (Abrupt shutdown or Graceful shutdown).
  - c Writing TCSR.PRESCALE to set TCNT count clock frequency.
  - d Setting TCNT, TDHR and TDFR.
  
- 2 Enable the clock.
  - a Writing TCSR.PWM\_EN to set whether enable PWM or disable PWM.
  - b Writing TCSR.EXT\_EN, TCSR.RTC\_EN or TCSR.PCK\_EN to 1 to select the input clock and enable the input clock. Only one of TCSR.EXT\_EN, TCSR.RTC\_EN and TCSR.PCK\_EN can be set to 1.

After initialize the register of timer, we should start the counter as follows:

- 3 Enable the counter.
 

Setting the TESR.TCST bit to 1 to enable the TCNT.

**NOTE:** The input clock and PCLK should follow the rules advanced before.

### 19.4.2 Disable and Shutdown Operation in TCU1 Mode

- 1 Setting the TECR.TCCL bit to 1 to disable the TCNT.

### 19.4.3 Basic Operation in TCU2 Mode

The value of TDFR should be bigger than TDHR, and the minimum settings are TDHR = 0 and TDFR = 1. In this case, the timer output clock cycle is the input clock  $\times 1/2$ . If TDHR > TDFR, no comparison

TFHR signal is generated.

Initial state is that TCSR.PRESCALE=0, TCSR.PWM\_EN=0 and TCENR=0.

- 1 Reset the TCU.
  - a Writing TCSR.PCK\_EN to 1 to select pclk as the input clock.
  - b Set TCSR.CLRZ to 1 to clear TCNT or set TCNT to an initial value.
  - c Writing TCSR.PCK\_EN to 0 to close the input clock.
- 2 Initial the configuration.
  - a Setting TDHR and TDFR.
  - b Writing TCSR.INITL to initialize PWM output level (if used PWM).
  - c Writing TCSR.PRESCALE to set TCNT count clock frequency.
  - d Writing TCSR.EXT\_EN, TCSR.RTC\_EN or TCSR.PCK\_EN to 1 to select the input clock and enable the input clock. Only one of TCSR.EXT\_EN, TCSR.RTC\_EN and TCSR.PCK\_EN can be set to 1.
  - e Writing TCSR.PWM\_EN to set whether enable PWM or disable PWM.

After initialize the register of timer, we should start the counter as follows:

- 3 Setting the TESR.TCST bit to 1 to enable the TCNT.

#### NOTES:

- 1 You can clear the counter when counter is working.
  - a Set TCSR.CLRZ to 1 to clear TCNT.
  - b Wait till TSTR.BUSY = 0, that is the counter have been cleared.
- 2 You can enable PWM or disable PWM the counter when counter is working.
  - a Set TCSR.PWM\_EN to 1 to enable PWM.
  - b Set TCSR.PWM\_EN to 0 to disable PWM.

#### 19.4.4 Disable and Shutdown Operation in TCU2 Mode

- 1 Writing TCSR.PWM\_EN to 0 to disable PWM.
- 2 Setting the TECR.TCCL bit to 1 to disable the TCNT.
- 3 Wait till TSTR.BUSY = 0, that is the reset of counter is finished.

#### 19.4.5 Read Counter in TCU2 Mode

If you want to read the data from register TCNT when the TCU is working, you can check TSTR.REAL whether it is good or not. It is suggested that:

- 1 If TSTR.REAL==1, the data read is available.
- 2 If TSTR.REAL==0, reread the counter till TSTR.REAL==1, the data read is available.
- 3 If TSTR.REAL is always 0, you can read some data, and lose some data that is quick different from the others. Then choose a data from them as the available data.

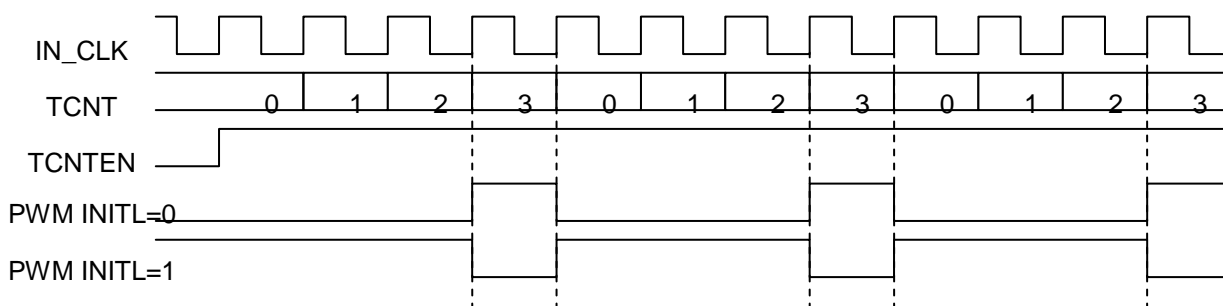
#### NOTES:

- 1 It suggested that (1), (2) is often used when the counter clock is very slow.
- 2 It suggested that (3) is often used when the counter clock is very fast.

#### 19.4.6 Pulse Width Modulator (PWM)

Timer 0~7 can be used as Pulse Width Modulator (PWM). The PWM can be used to control the back light inverter or adjust bright or contrast of LCD panel.

FULL comparison match signal and HALF comparison match signal can determine an attribute of the PWM\_OUT waveform. FULL comparison match signal specifies the clock cycle for the PWM module clock. HALF comparison match signal specifies the duty ratio for the PWM module clock.



#### 19.4.7 Trackball Input Waveform Detect

Timer 0, 3~7 can be used as a waveform edge counter to count both positive edge and negative edge of an external input waveform. For example, a trackball device's input. 4 timers will need to count all four directions (up, down, left, right). You need configure relate GPIO (set relate 4 PWM IO as input) and set relate TCSR.PWM\_IN\_EN to 1. Both relate TDFR and TDHR need to set to 0xFFFF, unless you need a special interrupt when the counter hit TDFR or TDHR. The counter will clear to 0 when hit TDFR.

Before the timer counter begin to count up, we need to do as follows:

- 1 Initial the configuration.
  - a Writing TCSR.SD to setting the shutdown mode (Abrupt shutdown or Graceful shutdown).
  - b Writing TCSR.PRESCALE to set to 0.
  - c Setting TCNT, TDHR and TDFR.
- 2 Enable the clock.
  - a Writing TCSR.PWM\_EN to disable PWM.
  - b Writing TCSR.EXT\_EN, TCSR.RTC\_EN and TCSR.PCK\_EN to 0, TCSR.PWM\_IN\_EN to 1 to select the input clock and enable the input clock.

After initialize the register of timer, we should start the counter as follows:

- 3 Enable the counter.
 

Setting the TESR.TCST bit to 1 to enable the TCNT.



**NOTE:** The input clock and PCLK should follow the rules advanced before.

### 19.4.8 Basic Operation in FIFO Mode 1

1. set TCSR, please keep PWM\_EN to 0.
2. set TCUMOD to 0x0000\_0001.
3. push data into fifo, through write data in TFWD. fifo's depth is 16, do not over flow.  
**do not push data like 0x0000\_XXXX, or 0xFFFF\_0000 X: nonzero value**
4. set the TCSR.PWM\_EN to 1.
5. set the TESR to enable the TCNT.
6. push some data into fifo is optional. Please ensure the TFIFOSR. **fifo\_w\_flag is 0, before push data into fifo. and do not over flow. do not push data like 0x0000\_XXXX, or 0xFFFF\_0000**
7. when TFR.fifo\_empty\_flag is 1, symbol the fifo is empty..
8. when fifo match finished ,if you want to do it again, please stop the TCNT count.
9. set TCSR,please keep PWM\_EN to 0.
10. set TFCR to clr the flag.
11. set TCUMOD.fifo\_clr to 1.
12. set TCUMOD.fifo\_clr to 0.
13. do step2 ~step7

```

example //write_reg(address,data)
1. write_reg(0x1000204C,0x00000002);
2. write_reg(0x10002100,0x00000001);
3. write_reg(0x10002104,0x00010001);
 write_reg(0x10002104,0x00020002);
 write_reg(0x10002104,0x00030003); //write fifo
4. apb_write_reg(0x1000204C,0x00000082);
5. apb_write_reg(0x10002014,0x00000001);
6. if((read_reg(0x10002108) & 0x1) != 1)
 {
 write_reg(0x10002104,0x00040004); //write fifo
 }

```

### 19.4.9 Basic Operation in FIFO Mode 2

1. set the TCSR, Please keep PWM\_EN to 0.
2. set TCUMOD to 0x0000\_0001.
3. push data into fifo, through write data in TFWD. fifo's depth is 16, do not over flow.  
**do not push data like 0x0000\_XXXX, or 0xFFFF\_0000**
4. set TCUMOD.
5. set the TCSR.PWM\_EN to 1.
6. set the TESR to enable the TCNT.
7. **if step 4 has been done, do not push data into fifo,**
8. when TFIFOSR. **cyc\_timer == TCUMOD. cyc\_num** , It symbol finished.

9. if you want do again, please stop the TCNT count
10. set TCUMOD.fifo\_clr to 1.
11. set TCUMOD.fifo\_clr to 0.
12. do step1 ~ step 8

example

1. write\_reg(0x1000204C,0x00000002);
2. write\_reg(0x10002100,0x00000001);
3. write\_reg(0x10002104,0x00010001);  
write\_reg(0x10002104,0x00020002);  
write\_reg(0x10002104,0x00030003);  
write\_reg(0x10002104,0x00040004);  
write\_reg(0x10002104,0x00050005);
4. write\_reg(0x10002100,0x00002A59);  
// conduct ten cycles, each cycle would match five data from fifo
5. write\_reg(0x1000204C,0x00000082);
6. write\_reg(0x10002014,0x00000001);
7. if(TFIFOSR. cyc\_timer == TCUMOD. cyc\_num)  
{  
finish  
}

## 20 Operating System Timer

### 20.1 Overview

The OST (Operating System Timer) contains one 64-bit programmable timer. It can be used as operating system timer.

OST has the following features:

- OST includes:
  - 64-bit Counter
  - 32-bit Compare Data Register
  - Control Register
- Independent clock for each counter, selectable by software
  - PCLK, EXTAL and RTCCLK can be used as the clock for counter
  - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software
- Match interrupt can be generated for OST using the compare data registers
  - Interrupt flag and interrupt mask is same with TCU in TCU spec

## 20.2 Register Description

In this section, we will describe the registers in OST. Following table lists all the registers definition. All OST register's 32bit address is physical address. And detailed function of each register will be described below.

| Name        | Description                                          | RW | Reset Value | Address    | Access Size |
|-------------|------------------------------------------------------|----|-------------|------------|-------------|
| OSTDR       | Operating System Timer Data Register                 | RW | 0x????????  | 0x100020e0 | 32          |
| OSTCNTL     | Operating System Timer Counter Lower 32 Bits         | RW | 0x????????  | 0x100020e4 | 32          |
| OSTCNTH     | Operating System Timer Counter Higher 32 Bits        | RW | 0x????????  | 0x100020e8 | 32          |
| OSTCSR      | Operating System Timer Control Register              | RW | 0x0000      | 0x100020ec | 16          |
| OSTCNTH BUF | Operating System Timer Counter Higher 32 Bits Buffer | R  | 0x????????  | 0x100020fc | 32          |

### 20.2.1 Operating System Control Register (OSTCSR)

The TCSR is a 16-bit read/write register. It contains the control bits for OST. It is initialized to 0x00 by any reset.

| OSTCSR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x100020ec |          |    |    |    |    |          |   |   |          |   |        |        |        |   |   |   |
|--------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------|----------|----|----|----|----|----------|---|---|----------|---|--------|--------|--------|---|---|---|
| Bit    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15         | 14       | 13 | 12 | 11 | 10 | 9        | 8 | 7 | 6        | 5 | 4      | 3      | 2      | 1 | 0 |   |
|        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CNT_MD     | Reserved |    |    |    | SD | Reserved |   |   | PRESCALE |   | EXT_EN | RTC_EN | PCK_EN |   |   |   |
| RST    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0          | 0        | 0  | 0  | 0  | 0  | 0        | 0 | 0 | 0        | 0 | 0      | 0      | 0      | 0 | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                           | RW |
|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 15   | CNT_MD   | Counter mode choose bit.<br>0: When the value counter is equal to compare value, the counter will be cleared, and increase from 0.<br>1: When the value counter is equal to compare value, the counter will go on increasing till overflow, and then increase from 0. |    |
| 14:6 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                  | R  |
| 9    | SD       | Shut Down (SD) the PWM output. It is only used in TCU1 mode.<br>0: Graceful shutdown (only used when CNT_MD = 0)<br>1: Abrupt shutdown                                                                                                                                | RW |
| 5:3  | PRESCALE | These bits select the TCNT count clock frequency.                                                                                                                                                                                                                     | RW |

|   |        | Bit 2                                                              | Bit1 | Bit 0 | Description              |    |  |
|---|--------|--------------------------------------------------------------------|------|-------|--------------------------|----|--|
|   |        | 0                                                                  | 0    | 0     | Internal clock: CLK/1    |    |  |
|   |        | 0                                                                  | 0    | 1     | Internal clock: CLK/4    |    |  |
|   |        | 0                                                                  | 1    | 0     | Internal clock: CLK/16   |    |  |
|   |        | 0                                                                  | 1    | 1     | Internal clock: CLK/64   |    |  |
|   |        | 1                                                                  | 0    | 0     | Internal clock: CLK/256  |    |  |
|   |        | 1                                                                  | 0    | 1     | Internal clock: CLK/1024 |    |  |
|   |        | 110~111                                                            |      |       | Reserved                 |    |  |
| 2 | EXT_EN | Select EXTAL as the timer clock input.<br>0: Disable<br>1: Enable  |      |       |                          | RW |  |
| 1 | RTC_EN | Select RTCCLK as the timer clock input.<br>1: Enable<br>0: Disable |      |       |                          | RW |  |
| 0 | PCK_EN | Select PCLK as the timer clock input.<br>1: Enable<br>0: Disable   |      |       |                          | RW |  |

**NOTE:** The input clock of timer and the PCLK should keep to the rules as follows.

| Input clock of timer: IN_CLK                                  | Clock generated from the frequency divider (PRESCALE): DIV_CLK |
|---------------------------------------------------------------|----------------------------------------------------------------|
| PCK_EN == 0, RTC_EN == 1 and EXT_EN == 0<br>(IN_CLK = RTCCLK) | $f_{DIV\_CLK} < \frac{1}{2} f_{PCLK}$                          |
| PCK_EN == 0, RTC_EN == 0 and EXT_EN == 1<br>(IN_CLK = EXTAL)  | $f_{DIV\_CLK} < \frac{1}{2} f_{PCLK}$                          |
| PCK_EN == 1, RTC_EN == 0 and EXT_EN == 0<br>(IN_CLK = PCLK)   | ANY                                                            |

### 20.2.2 Operating System Timer Data Register (OSTDR)

The operating system timer data register OSTDR is used to store the data to be compared with the content of the operating system timer up-counter OSTCNT. This register can be directly read and written. Please also refer to CNT\_MD bit of register **OSTCSR** to understand the counter behavior. (Default: indeterminate)

| OSTDR |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x100020e0 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31    | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | OSTDR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST   | ?     | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?          | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

### 20.2.3 Operating System Timer Counter (OSTCNTH, OSTCNTL)

The operating system timer counter (OSTCNT) is a 64-bit read/write counter. The up-counter OSTCNT can be set by software and counts up using the prescaler output clock. The data can be read out at any time. The counter data can be written at any time. (Default: indeterminate)

|                |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|----------------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|
| <b>OSTCNTH</b> |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x100020e8</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>     | 31      | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |
|                | OSTCNTH |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>     | ?       | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                 | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

|                |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|----------------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|
| <b>OSTCNTL</b> |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x100020e4</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>     | 31      | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |
|                | OSTCNTL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>     | ?       | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                 | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

### 20.2.4 Operating System Timer Counter high 32 bits buffer (OSTCNTHBUF)

The operating system timer counter high 32 bits buffer OSTCNTHBUF is used to store the high 32 bits of OSTCNT when its lower 32 bits are read by software. This register can be directly read. (Default: indeterminate)

|                   |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
|-------------------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|
| <b>OSTCNTHBUF</b> |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x100020fc</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>        | 31         | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |
|                   | OSTCNTHBUF |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>        | ?          | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                 | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

## 20.3 Operation

### 20.3.1 Basic Operation

Before the timer counter begin to count up, we need to do as follows:

- 1 Initial the configuration.
  - a Writing TCSR.SD to setting the shutdown mode (Abrupt shutdown or Graceful shutdown).
  - b Writing OSTCSR.PRESCALE to set OSTCNT count clock frequency.
  - c Setting OSTCNTL/H and OSTDR.

- 2 Enable the clock.

Writing OSTCSR.EXT\_EN, OSTCSR.RTC\_EN or OSTCSR.PCK\_EN to 1 to select the input clock and enable the input clock. Only one of OSTCSR.EXT\_EN, OSTCSR.RTC\_EN and OSTCSR.PCK\_EN can be set to 1.

After initialize the register of timer, we should start the counter as follows:

- 3 Enable the counter.

Setting the TESR.OSTCST bit to 1 to enable the OSTCNT.

**NOTE:** The input clock and PCLK should follow the rules advanced before.

### 20.3.2 Disable and Shutdown Operation

- 1 Setting the TECR.OSTCCL bit to 1 to disable the OSTCNT.

## 21 Interrupt Controller

### 21.1 Overview

This chapter describes the interrupt controller included in the XBurst Processor, explains its modes of operation, and defines its registers. The interrupt controller controls the interrupt sources available to the processor and contains the location of the interrupt source to allow software to determine source of all interrupts. It also determines whether an IRQ occurs or not and masks the interrupts.

Features:

- Total 64 interrupt sources
- Each interrupt source can be independently enabled
- Priority mechanism to indicate highest priority interrupt
- All the registers are accessed by CPU
- Unmasked interrupts can wake up the chip in sleep mode
- Another set of source, mask and pending registers to serve for PDMA

### 21.2 Register Description

Table 21-1 lists the registers of Interrupt Controller. All of these registers are 32bit, and each bit of the register represents or controls one interrupt source that list in Table 21-1.

**Table 21-1 INTC Register**

| Name   | Description                    | RW | Reset Value | Address    | Access Size |
|--------|--------------------------------|----|-------------|------------|-------------|
| ICSR0  | IRQ Source Register            | R  | 0x00000000  | 0x10001000 | 32          |
| ICMR0  | IRQ Mask Register              | RW | 0xFFFFFFFF  | 0x10001004 | 32          |
| ICMSR0 | IRQ Mask Set Register          | W  | 0x????????  | 0x10001008 | 32          |
| ICMCR0 | IRQ Mask Clear Register        | W  | 0x????????  | 0x1000100C | 32          |
| ICPR0  | IRQ Pending Register           | R  | 0x00000000  | 0x10001010 | 32          |
| ICSR1  | IRQ Source Register            | R  | 0x00000000  | 0x10001020 | 32          |
| ICMR1  | IRQ Mask Register              | RW | 0xFFFFFFFF  | 0x10001024 | 32          |
| ICMSR1 | IRQ Mask Set Register          | W  | 0x????????  | 0x10001028 | 32          |
| ICMCR1 | IRQ Mask Clear Register        | W  | 0x????????  | 0x1000102C | 32          |
| ICPR1  | IRQ Pending Register           | R  | 0x00000000  | 0x10001030 | 32          |
| DSR0   | IRQ Source Register0 for PDMA  | R  | 0x00000000  | 0x10001034 | 32          |
| DMR0   | IRQ Mask Register0 for PDMA    | RW | 0xFFFFFFFF  | 0x10001038 | 32          |
| DPR0   | IRQ Pending Register0 for PDMA | R  | 0x00000000  | 0x1000103C | 32          |
| DSR1   | IRQ Source Register1 for PDMA  | R  | 0x00000000  | 0x10001040 | 32          |
| DMR1   | IRQ Mask Register1 for PDMA    | RW | 0xFFFFFFFF  | 0x10001044 | 32          |
| DPR1   | IRQ Pending Register1 for PDMA | R  | 0x00000000  | 0x10001048 | 32          |



### 21.2.1 Interrupt Controller Source Register (ICSR0)

This register contains all the interrupts' status. A "1" indicates that the corresponding interrupt is pending . A "0" indicates that the interrupt is not pending now. The register is read only.

| ICSR0 |     |     |     |     |      |      |      |         |      |      |     |      |     |      |       | 0x10001000 |       |       |       |       |      |      |       |      |      |          |      |             |      |     |      |      |
|-------|-----|-----|-----|-----|------|------|------|---------|------|------|-----|------|-----|------|-------|------------|-------|-------|-------|-------|------|------|-------|------|------|----------|------|-------------|------|-----|------|------|
| Bit   | 31  | 30  | 29  | 28  | 27   | 26   | 25   | 24      | 23   | 22   | 21  | 20   | 19  | 18   | 17    | 16         | 15    | 14    | 13    | 12    | 11   | 10   | 9     | 8    | 7    | 6        | 5    | 4           | 3    | 2   | 1    | 0    |
|       | LCD | CIM | IPU | GPS | TCU0 | TCU1 | TCU2 | GPS_1MS | LCD1 | IPU1 | OTG | EHCI | X2D | SADC | GPIO0 | GPIO1      | GPIO2 | GPIO3 | GPIO4 | GPIO5 | TSS1 | PDMA | TSS10 | SSI0 | SSI1 | Reserved | OHCI | HDMI_WAKEUP | HDMI | BCH | AIC0 | AIC1 |
| RST   | 0   | 0   | 0   | 0   | 0    | 0    | 0    | 0       | 0    | 0    | 0   | 0    | 0   | 0    | 0     | 0          | 0     | 0     | 0     | 0     | 0    | 0    | 0     | 0    | 0    | 0        | 0    | 0           | 0    | 0   | 0    | 0    |

| Bits Of ICSR0 | Description                                        |
|---------------|----------------------------------------------------|
| 0             | The corresponding interrupt source is not pending. |
| 1             | The corresponding interrupt source is pending.     |

### 21.2.2 Interrupt Controller Source Register (ICSR1)

This register contains all the interrupts' status. A "1" indicates that the corresponding interrupt is pending . A "0" indicates that the interrupt is not pending now. The register is read only.

| ICSR1 |     |     |       |      |      |      |      |      |      |      |          |     |       |       |       | 0x10001020 |     |       |          |       |          |       |     |      |          |     |      |      |      |       |     |     |
|-------|-----|-----|-------|------|------|------|------|------|------|------|----------|-----|-------|-------|-------|------------|-----|-------|----------|-------|----------|-------|-----|------|----------|-----|------|------|------|-------|-----|-----|
| Bit   | 31  | 30  | 29    | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21       | 20  | 19    | 18    | 17    | 16         | 15  | 14    | 13       | 12    | 11       | 10    | 9   | 8    | 7        | 6   | 5    | 4    | 3    | 2     | 1   | 0   |
|       | GPU | VPU | PDMAM | SMB0 | SMB1 | SMB2 | SMB3 | SMB4 | ETHC | NEMC | Reserved | DDR | UART0 | UART1 | UART2 | UART3      | CPM | HARB0 | Reserved | HARB2 | Compress | GPVLC | KBC | PCM0 | Reserved | SCC | MSC0 | MSC1 | MSC2 | URAT4 | OWI | RTC |
| RST   | 0   | 0   | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0        | 0   | 0     | 0     | 0     | 0          | 0   | 0     | 0        | 0     | 0        | 0     | 0   | 0    | 0        | 0   | 0    | 0    | 0    | 0     | 0   | 0   |

| Bits Of ICSR1 | Description                                        |
|---------------|----------------------------------------------------|
| 0             | The corresponding interrupt source is not pending. |
| 1             | The corresponding interrupt source is pending.     |

### 21.2.3 Interrupt Controller Mask Register (ICMR0)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. Its value can be changed either by writing ICMSR and ICMCR or by writing itself. The masked interrupts are invisible to the processor.

| ICMR0 |    |     |     |     |     |      |      |      |         |      |      |     |      |     |      | 0x10001004 |       |       |       |       |       |      |      |       |      |      |          |      |             |      |     |      |      |
|-------|----|-----|-----|-----|-----|------|------|------|---------|------|------|-----|------|-----|------|------------|-------|-------|-------|-------|-------|------|------|-------|------|------|----------|------|-------------|------|-----|------|------|
| Bit   | 31 | 30  | 29  | 28  | 27  | 26   | 25   | 24   | 23      | 22   | 21   | 20  | 19   | 18  | 17   | 16         | 15    | 14    | 13    | 12    | 11    | 10   | 9    | 8     | 7    | 6    | 5        | 4    | 3           | 2    | 1   | 0    |      |
|       |    | LCD | CIM | IPU | GPS | TCU0 | TCU1 | TCU2 | GPS_1MS | LCD1 | IPU1 | OTG | EHCI | X2D | SADC | GPIO0      | GPIO1 | GPIO2 | GPIO3 | GPIO4 | GPIO5 | TSS1 | PDMA | TSS10 | SSI0 | SSI1 | Reserved | OHCI | HDMI_WAKEUP | HDMI | BCH | AIC0 | AIC1 |
| RST   | 1  | 1   | 1   | 1   | 1   | 1    | 1    | 1    | 1       | 1    | 1    | 1   | 1    | 1   | 1    | 1          | 1     | 1     | 1     | 1     | 1     | 1    | 1    | 1     | 1    | 1    | 1        | 1    | 1           | 1    | 1   | 1    |      |

| Bits Of ICMR0 | Description                                |
|---------------|--------------------------------------------|
| 0             | The corresponding interrupt is not masked. |
| 1             | The corresponding interrupt is masked.     |

### 21.2.4 Interrupt Controller Mask Register (ICMR1)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. Its value can be changed either by writing ICMSR and ICMCR or by writing itself. The masked interrupts are invisible to the processor.

| ICMR1 |    |     |     |       |      |      |      |      |      |      |      |          |     |       |       | 0x10001024 |       |     |       |          |       |          |       |     |      |          |     |      |      |      |       |     |     |
|-------|----|-----|-----|-------|------|------|------|------|------|------|------|----------|-----|-------|-------|------------|-------|-----|-------|----------|-------|----------|-------|-----|------|----------|-----|------|------|------|-------|-----|-----|
| Bit   | 31 | 30  | 29  | 28    | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20       | 19  | 18    | 17    | 16         | 15    | 14  | 13    | 12       | 11    | 10       | 9     | 8   | 7    | 6        | 5   | 4    | 3    | 2    | 1     | 0   |     |
|       |    | GPU | VPU | PDMAM | SMB0 | SMB1 | SMB2 | SMB3 | SMB4 | ETHC | NEMC | Reserved | DDR | UART0 | UART1 | UART2      | UART3 | CPM | HARB0 | Reserved | HARB2 | Compress | GPVLC | KBC | PCM0 | Reserved | SCC | MSC0 | MSC1 | MSC2 | URAT4 | OWI | RTC |
| RST   | 1  | 1   | 1   | 1     | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1        | 1   | 1     | 1     | 1          | 1     | 1   | 1     | 1        | 1     | 1        | 1     | 1   | 1    | 1        | 1   | 1    | 1    | 1    | 1     | 1   |     |

| Bits Of ICMR1 | Description                                |
|---------------|--------------------------------------------|
| 0             | The corresponding interrupt is not masked. |
| 1             | The corresponding interrupt is masked.     |

### 21.2.5 Interrupt Controller Mask Set Register (ICMSR0)

This register is used to set bits in the interrupt mask register. This register is write only.

| ICMSR0 |    |     |     |     |     |      |      |      |         |      |      |     |      |     |      | 0x10001008 |       |       |       |       |       |      |      |       |      |      |          |      |             |      |     |      |      |
|--------|----|-----|-----|-----|-----|------|------|------|---------|------|------|-----|------|-----|------|------------|-------|-------|-------|-------|-------|------|------|-------|------|------|----------|------|-------------|------|-----|------|------|
| Bit    | 31 | 30  | 29  | 28  | 27  | 26   | 25   | 24   | 23      | 22   | 21   | 20  | 19   | 18  | 17   | 16         | 15    | 14    | 13    | 12    | 11    | 10   | 9    | 8     | 7    | 6    | 5        | 4    | 3           | 2    | 1   | 0    |      |
|        |    | LCD | CIM | IPU | GPS | TCU0 | TCU1 | TCU2 | GPS_1MS | LCD1 | IPU1 | OTG | EHCI | X2D | SADC | GPIO0      | GPIO1 | GPIO2 | GPIO3 | GPIO4 | GPIO5 | TSS1 | PDMA | TSS10 | SSI0 | SSI1 | Reserved | OHCI | HDMI_WAKEUP | HDMI | BCH | AIC0 | AIC1 |
| RST    | 0  | 0   | 0   | 0   | 0   | 0    | 0    | 0    | 0       | 0    | 0    | 0   | 0    | 0   | 0    | 0          | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0     | 0    | 0    | 0        | 0    | 0           | 0    | 0   | 0    |      |

| Bits Of ICMSR0 | Description                                    |
|----------------|------------------------------------------------|
| 0              | Ignore.                                        |
| 1              | Will set the corresponding interrupt mask bit. |

### 21.2.6 Interrupt Controller Mask Set Register (ICMSR1)

This register is used to set bits in the interrupt mask register. This register is write only.

| ICMSR1 |     |     |       |      |      |      |      |      |      |      |          |     |       |       |       | 0x10001028 |     |       |          |       |          |       |     |      |          |     |      |      |      |       |     |     |
|--------|-----|-----|-------|------|------|------|------|------|------|------|----------|-----|-------|-------|-------|------------|-----|-------|----------|-------|----------|-------|-----|------|----------|-----|------|------|------|-------|-----|-----|
| Bit    | 31  | 30  | 29    | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21       | 20  | 19    | 18    | 17    | 16         | 15  | 14    | 13       | 12    | 11       | 10    | 9   | 8    | 7        | 6   | 5    | 4    | 3    | 2     | 1   | 0   |
|        | GPU | VPU | PDMAM | SMB0 | SMB1 | SMB2 | SMB3 | SMB4 | ETHC | NEMC | Reserved | DDR | UART0 | UART1 | UART2 | UART3      | CPM | HARB0 | Reserved | HARB2 | Compress | GPVLC | KBC | PCMO | Reserved | SCC | MSC0 | MSC1 | MSC2 | URAT4 | OWI | RTC |
| RST    | 0   | 0   | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0        | 0   | 0     | 0     | 0     | 0          | 0   | 0     | 0        | 0     | 0        | 0     | 0   | 0    | 0        | 0   | 0    | 0    | 0    | 0     | 0   | 0   |

| Bits Of ICMSR1 | Description                                    |
|----------------|------------------------------------------------|
| 0              | Ignore.                                        |
| 1              | Will set the corresponding interrupt mask bit. |

### 21.2.7 Interrupt Controller Mask Clear Register (ICMCR0)

This register is used to clear bits in the interrupt mask register. This register is write only.

| ICMCR0 |     |     |     |     |      |      |      |         |      |      |     |      |     |      |       | 0x1000100C |       |       |       |       |       |      |       |      |      |          |      |             |      |     |      |      |
|--------|-----|-----|-----|-----|------|------|------|---------|------|------|-----|------|-----|------|-------|------------|-------|-------|-------|-------|-------|------|-------|------|------|----------|------|-------------|------|-----|------|------|
| Bit    | 31  | 30  | 29  | 28  | 27   | 26   | 25   | 24      | 23   | 22   | 21  | 20   | 19  | 18   | 17    | 16         | 15    | 14    | 13    | 12    | 11    | 10   | 9     | 8    | 7    | 6        | 5    | 4           | 3    | 2   | 1    | 0    |
|        | LCD | CIM | IPU | GPS | TCU0 | TCU1 | TCU2 | GPS_1MS | LCD1 | IPU1 | OTG | EHCI | X2D | SADC | GPIO0 | GPIO1      | GPIO2 | GPIO3 | GPIO4 | GPIO5 | TSS11 | PDMA | TSS10 | SSI0 | SSI1 | Reserved | OHCI | HDMI_WAKEUP | HDMI | BCH | AIC0 | AIC1 |
| RST    | 0   | 0   | 0   | 0   | 0    | 0    | 0    | 0       | 0    | 0    | 0   | 0    | 0   | 0    | 0     | 0          | 0     | 0     | 0     | 0     | 0     | 0    | 0     | 0    | 0    | 0        | 0    | 0           | 0    | 0   | 0    | 0    |

| Bits Of ICMCR0 | Description                                      |
|----------------|--------------------------------------------------|
| 0              | Ignore.                                          |
| 1              | Will clear the corresponding interrupt mask bit. |

### 21.2.8 Interrupt Controller Mask Clear Register (ICMCR1)

This register is used to clear bits in the interrupt mask register. This register is write only.

| ICMCR1 |     |     |       |      |      |      |      |      |      |      |          |     |       |       |       | 0x1000102C |     |       |          |       |          |       |     |      |          |     |      |      |      |       |     |     |
|--------|-----|-----|-------|------|------|------|------|------|------|------|----------|-----|-------|-------|-------|------------|-----|-------|----------|-------|----------|-------|-----|------|----------|-----|------|------|------|-------|-----|-----|
| Bit    | 31  | 30  | 29    | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21       | 20  | 19    | 18    | 17    | 16         | 15  | 14    | 13       | 12    | 11       | 10    | 9   | 8    | 7        | 6   | 5    | 4    | 3    | 2     | 1   | 0   |
|        | GPU | VPU | PDMAM | SMB0 | SMB1 | SMB2 | SMB3 | SMB4 | ETHC | NEMC | Reserved | DDR | UART0 | UART1 | UART2 | UART3      | CPM | HARB0 | Reserved | HARB2 | Compress | GPVLC | KBC | PCM0 | Reserved | SCC | MSC0 | MSC1 | MSC2 | URAT4 | OWI | RTC |
| RST    | 0   | 0   | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0        | 0   | 0     | 0     | 0     | 0          | 0   | 0     | 0        | 0     | 0        | 0     | 0   | 0    | 0        | 0   | 0    | 0    | 0    | 0     | 0   | 0   |

| Bits Of ICMCR1 | Description                                      |
|----------------|--------------------------------------------------|
| 0              | Ignore.                                          |
| 1              | Will clear the corresponding interrupt mask bit. |

### 21.2.9 Interrupt Controller Pending Register (ICPR0)

This register contains the status of the interrupt sources after masking. This register is read only.

| ICPR0 |     |     |     |     |      |      |      |         |      |      |     |      |     |      |       | 0x10001010 |       |       |       |       |      |      |       |      |      |          |      |             |      |     |      |      |
|-------|-----|-----|-----|-----|------|------|------|---------|------|------|-----|------|-----|------|-------|------------|-------|-------|-------|-------|------|------|-------|------|------|----------|------|-------------|------|-----|------|------|
| Bit   | 31  | 30  | 29  | 28  | 27   | 26   | 25   | 24      | 23   | 22   | 21  | 20   | 19  | 18   | 17    | 16         | 15    | 14    | 13    | 12    | 11   | 10   | 9     | 8    | 7    | 6        | 5    | 4           | 3    | 2   | 1    | 0    |
|       | LCD | CIM | IPU | GPS | TCU0 | TCU1 | TCU2 | GPS_1MS | LCD1 | IPU1 | OTG | EHCI | X2D | SADC | GPIO0 | GPIO1      | GPIO2 | GPIO3 | GPIO4 | GPIO5 | TSS1 | PDMA | TSS10 | SSIO | SSI1 | Reserved | OHCI | HDMI_WAKEUP | HDMI | BCH | AIC0 | AIC1 |
| RST   | 0   | 0   | 0   | 0   | 0    | 0    | 0    | 0       | 0    | 0    | 0   | 0    | 0   | 0    | 0     | 0          | 0     | 0     | 0     | 0     | 0    | 0    | 0     | 0    | 0    | 0        | 0    | 0           | 0    | 0   | 0    | 0    |

| Bits Of ICPR0 | Description                                                               |
|---------------|---------------------------------------------------------------------------|
| 0             | The corresponding interrupt is not active or is masked.                   |
| 1             | The corresponding interrupt is active and is not masked to the processor. |

### 21.2.10 Interrupt Controller Pending Register (ICPR1)

This register contains the status of the interrupt sources after masking. This register is read only.

| ICPR1 |     |     |       |      |      |      |      |      |      |      |          |     |       |       |       | 0x10001030 |     |       |          |       |          |       |     |      |          |     |      |      |      |       |     |     |
|-------|-----|-----|-------|------|------|------|------|------|------|------|----------|-----|-------|-------|-------|------------|-----|-------|----------|-------|----------|-------|-----|------|----------|-----|------|------|------|-------|-----|-----|
| Bit   | 31  | 30  | 29    | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21       | 20  | 19    | 18    | 17    | 16         | 15  | 14    | 13       | 12    | 11       | 10    | 9   | 8    | 7        | 6   | 5    | 4    | 3    | 2     | 1   | 0   |
|       | GPU | VPU | PDMAM | SMB0 | SMB1 | SMB2 | SMB3 | SMB4 | ETHC | NEMC | Reserved | DDR | UART0 | UART1 | UART2 | UART3      | CPM | HARB0 | Reserved | HARB2 | Compress | GPVLC | KBC | PCM0 | Reserved | SCC | MSC0 | MSC1 | MSC2 | URAT4 | OWI | RTC |
| RST   | 0   | 0   | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0        | 0   | 0     | 0     | 0     | 0          | 0   | 0     | 0        | 0     | 0        | 0     | 0   | 0    | 0        | 0   | 0    | 0    | 0    | 0     | 0   | 0   |

| Bits Of ICPR1 | Description                                                               |
|---------------|---------------------------------------------------------------------------|
| 0             | The corresponding interrupt is not active or is masked.                   |
| 1             | The corresponding interrupt is active and is not masked to the processor. |

### 21.2.11 Interrupt Source Register0 for PDMA (DSR0)

This register contains status of all interrupts. A “1” indicates the corresponding interrupt is pending. The register is read only.

| DSR0 |     |     |     |     |      |      |      |         |      |      |     |      |     |      |       | 0x10001034 |       |       |       |       |      |      |       |      |      |          |      |             |      |     |      |      |
|------|-----|-----|-----|-----|------|------|------|---------|------|------|-----|------|-----|------|-------|------------|-------|-------|-------|-------|------|------|-------|------|------|----------|------|-------------|------|-----|------|------|
| Bit  | 31  | 30  | 29  | 28  | 27   | 26   | 25   | 24      | 23   | 22   | 21  | 20   | 19  | 18   | 17    | 16         | 15    | 14    | 13    | 12    | 11   | 10   | 9     | 8    | 7    | 6        | 5    | 4           | 3    | 2   | 1    | 0    |
|      | LCD | CIM | IPU | GPS | TCU0 | TCU1 | TCU2 | GPS_1MS | LCD1 | IPU1 | OTG | EHCI | X2D | SADC | GPIO0 | GPIO1      | GPIO2 | GPIO3 | GPIO4 | GPIO5 | TSS1 | PDMA | TSS10 | SS10 | SS11 | Reserved | OHCI | HDMI_WAKEUP | HDMI | BCH | AIC0 | AIC1 |
| RST  | 0   | 0   | 0   | 0   | 0    | 0    | 0    | 0       | 0    | 0    | 0   | 0    | 0   | 0    | 0     | 0          | 0     | 0     | 0     | 0     | 0    | 0    | 0     | 0    | 0    | 0        | 0    | 0           | 0    | 0   | 0    | 0    |

| Bits Of DSR0 | Description                                        |
|--------------|----------------------------------------------------|
| 0            | The corresponding interrupt source is not pending. |
| 1            | The corresponding interrupt source is pending.     |

### 21.2.12 Interrupt Mask Register0 for PDMA (DMR0)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. The masked interrupts are invisible to the processor.

| DMR0 |     |     |     |     |      |      |      |         |      |      |     |      |     |      |       | 0x10001038 |       |       |       |       |      |      |       |      |      |          |      |             |      |     |      |      |
|------|-----|-----|-----|-----|------|------|------|---------|------|------|-----|------|-----|------|-------|------------|-------|-------|-------|-------|------|------|-------|------|------|----------|------|-------------|------|-----|------|------|
| Bit  | 31  | 30  | 29  | 28  | 27   | 26   | 25   | 24      | 23   | 22   | 21  | 20   | 19  | 18   | 17    | 16         | 15    | 14    | 13    | 12    | 11   | 10   | 9     | 8    | 7    | 6        | 5    | 4           | 3    | 2   | 1    | 0    |
|      | LCD | CIM | IPU | GPS | TCU0 | TCU1 | TCU2 | GPS_1MS | LCD1 | IPU1 | OTG | EHCI | X2D | SADC | GPIO0 | GPIO1      | GPIO2 | GPIO3 | GPIO4 | GPIO5 | TSS1 | PDMA | TSS10 | SS10 | SS11 | Reserved | OHCI | HDMI_WAKEUP | HDMI | BCH | AIC0 | AIC1 |
| RST  | 1   | 1   | 1   | 1   | 1    | 1    | 1    | 1       | 1    | 1    | 1   | 1    | 1   | 1    | 1     | 1          | 1     | 1     | 1     | 1     | 1    | 1    | 1     | 1    | 1    | 1        | 1    | 1           | 1    | 1   | 1    | 1    |

| Bits Of DMR0 | Description                                        |
|--------------|----------------------------------------------------|
| 0            | The corresponding interrupt source is not pending. |
| 1            | The corresponding interrupt source is pending.     |

### 21.2.13 Interrupt Pending Register0 for PDMA (DPR0)

This register contains the status of the interrupt sources after masking. This register is read only.

| DPR0 |    |     |     |     |     |      |      |      |         |      |      |     |      |     |      | 0x1000103C |       |       |       |       |       |      |      |       |      |      |          |      |             |      |     |      |      |
|------|----|-----|-----|-----|-----|------|------|------|---------|------|------|-----|------|-----|------|------------|-------|-------|-------|-------|-------|------|------|-------|------|------|----------|------|-------------|------|-----|------|------|
| Bit  | 31 | 30  | 29  | 28  | 27  | 26   | 25   | 24   | 23      | 22   | 21   | 20  | 19   | 18  | 17   | 16         | 15    | 14    | 13    | 12    | 11    | 10   | 9    | 8     | 7    | 6    | 5        | 4    | 3           | 2    | 1   | 0    |      |
|      |    | LCD | CIM | IPU | GPS | TCU0 | TCU1 | TCU2 | GPS_1MS | LCD1 | IPU1 | OTG | EHCI | X2D | SADC | GPIO0      | GPIO1 | GPIO2 | GPIO3 | GPIO4 | GPIO5 | TSS1 | PDMA | TSS10 | SSIO | SS11 | Reserved | OHC1 | HDMI_WAKEUP | HDMI | BCH | AIC0 | AIC1 |
| RST  | 0  | 0   | 0   | 0   | 0   | 0    | 0    | 0    | 0       | 0    | 0    | 0   | 0    | 0   | 0    | 0          | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0     | 0    | 0    | 0        | 0    | 0           | 0    | 0   | 0    | 0    |

| Bits Of DPR0 | Description                                                               |
|--------------|---------------------------------------------------------------------------|
| 0            | The corresponding interrupt is not active or is masked.                   |
| 1            | The corresponding interrupt is active and is not masked to the processor. |

### 21.2.14 Interrupt Source Register1 to PDMA (DSR1)

This register contains status of all interrupts. A “1” indicates the corresponding interrupt is pending. The register is read only.

| DSR1 |    |     |     |          |      |      |      |      |      |      |      |          |     |       |       | 0x10001040 |       |     |       |          |       |          |       |     |      |          |     |      |      |      |       |     |     |
|------|----|-----|-----|----------|------|------|------|------|------|------|------|----------|-----|-------|-------|------------|-------|-----|-------|----------|-------|----------|-------|-----|------|----------|-----|------|------|------|-------|-----|-----|
| Bit  | 31 | 30  | 29  | 28       | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20       | 19  | 18    | 17    | 16         | 15    | 14  | 13    | 12       | 11    | 10       | 9     | 8   | 7    | 6        | 5   | 4    | 3    | 2    | 1     | 0   |     |
|      |    | GPU | VPU | Reserved | SMB0 | SMB1 | SMB2 | SMB3 | SMB4 | ETHC | NEMC | Reserved | DDR | UART0 | UART1 | UART2      | UART3 | CPM | HARB0 | Reserved | HARB2 | Compress | GPVLC | KBC | PCM0 | Reserved | SCC | MSC0 | MSC1 | MSC2 | URAT4 | OWI | RTC |
| RST  | 0  | 0   | 0   | 0        | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0        | 0   | 0     | 0     | 0          | 0     | 0   | 0     | 0        | 0     | 0        | 0     | 0   | 0    | 0        | 0   | 0    | 0    | 0    | 0     | 0   | 0   |

| Bits Of DSR1 | Description                                        |
|--------------|----------------------------------------------------|
| 0            | The corresponding interrupt source is not pending. |
| 1            | The corresponding interrupt source is pending.     |

### 21.2.15 Interrupt Mask Register1 for PDMA (DMR1)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. The masked interrupts are invisible to the processor.

| DMR1 |    |     |     |          |      |      |      |      |      |      |      |          |     |       |       | 0x10001044 |       |     |       |          |       |          |       |     |      |          |     |      |      |      |       |     |     |
|------|----|-----|-----|----------|------|------|------|------|------|------|------|----------|-----|-------|-------|------------|-------|-----|-------|----------|-------|----------|-------|-----|------|----------|-----|------|------|------|-------|-----|-----|
| Bit  | 31 | 30  | 29  | 28       | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20       | 19  | 18    | 17    | 16         | 15    | 14  | 13    | 12       | 11    | 10       | 9     | 8   | 7    | 6        | 5   | 4    | 3    | 2    | 1     | 0   |     |
|      |    | GPU | VPU | Reserved | SMB0 | SMB1 | SMB2 | SMB3 | SMB4 | ETHC | NEMC | Reserved | DDR | UART0 | UART1 | UART2      | UART3 | CPM | HARB0 | Reserved | HARB2 | Compress | GPVLC | KBC | PCM0 | Reserved | SCC | MSC0 | MSC1 | MSC2 | URAT4 | OWI | RTC |
| RST  | 1  | 1   | 1   | 1        | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1        | 1   | 1     | 1     | 1          | 1     | 1   | 1     | 1        | 1     | 1        | 1     | 1   | 1    | 1        | 1   | 1    | 1    | 1    | 1     | 1   | 1   |

| Bits Of DMR1 | Description                                |
|--------------|--------------------------------------------|
| 0            | The corresponding interrupt is not masked. |
| 1            | The corresponding interrupt is masked.     |

### 21.2.16 Interrupt Pending Register1 for PDMA (DPR1)

This register contains the status of the interrupt sources after masking. This register is read only.

| DPR1 |     |     |          |      |      |      |      |      |      |      |          |     |       |       |       | 0x10001048 |     |       |          |       |          |       |     |      |          |     |      |      |      |       |     |     |
|------|-----|-----|----------|------|------|------|------|------|------|------|----------|-----|-------|-------|-------|------------|-----|-------|----------|-------|----------|-------|-----|------|----------|-----|------|------|------|-------|-----|-----|
| Bit  | 31  | 30  | 29       | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21       | 20  | 19    | 18    | 17    | 16         | 15  | 14    | 13       | 12    | 11       | 10    | 9   | 8    | 7        | 6   | 5    | 4    | 3    | 2     | 1   | 0   |
|      | GPU | VPU | Reserved | SMB0 | SMB1 | SMB2 | SMB3 | SMB4 | ETHC | NEMC | Reserved | DDR | UART0 | UART1 | UART2 | UART3      | CPM | HARB0 | Reserved | HARB2 | Compress | GPVLC | KBC | PCM0 | Reserved | SCC | MSC0 | MSC1 | MSC2 | URAT4 | OWI | RTC |
| RST  | 0   | 0   | 0        | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0        | 0   | 0     | 0     | 0     | 0          | 0   | 0     | 0        | 0     | 0        | 0     | 0   | 0    | 0        | 0   | 0    | 0    | 0    | 0     | 0   | 0   |

| Bits Of ICPR0 | Description                                                               |
|---------------|---------------------------------------------------------------------------|
| 0             | The corresponding interrupt is not active or is masked.                   |
| 1             | The corresponding interrupt is active and is not masked to the processor. |

### 21.3 Software Considerations

The interrupt controller is reflecting the status of interrupts sources in the peripheral.

Software should perform the task - determine the interrupt source from in ICPRx. In this chip, pending interrupts have two levels in structure. Interrupting module in the system that contains more than one interrupt sources need software to determine how to service it by reading interrupt status registers within it.

In the interrupt handler, the serviced interrupt source needs to be cleared in the interrupting device. In order to make certain the cleared source request status has been reflected at the corresponding ICPRx bit, software should wait enough time before exiting interrupt state.

The procedure is described following:

- 1 Interrupt generated.
- 2 CPU query interrupt sources, saves the current environment and then goes to interrupt common service routine.
- 3 Get ICPRx.
- 4 Find the highest priority interrupt and vector it. (The software decides which one has the highest priority)
- 5 Mask the chosen interrupt by writing the register ICMSRx.
- 6 Enable the system interrupt to allow the interrupt nesting. (software decided)
- 7 Execute the interrupt handler and unmask it by writing the register ICMCRx when exit the

handler.

- 8 CPU restores the saved environment and exits the interrupt state.



## 22 Watchdog Timer

### 22.1 Overview

The watchdog timer is used to resume the processor whenever it is disturbed by malfunctions such as noise and system errors. The watchdog timer can generate the reset signal.

Features:

- Generates WDT reset
- A 16-bit Data register and a 16-bit counter
- Counter clock uses the input clock selected by software
  - PCLK, EXTAL and RTCCLK can be used as the clock for counter
  - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software

### 22.2 Register Description

In this section, we will describe the registers in WDT. Following table lists all the registers definition. All WDT register's 32bit address is physical address. And detailed function of each register will be described below.

| Name | Description                      | RW | Reset Value | Address    | Access Size |
|------|----------------------------------|----|-------------|------------|-------------|
| TDR  | Watchdog Timer Data Register     | RW | 0x????      | 0x10002000 | 16          |
| TCER | Watchdog Counter Enable Register | RW | 0x00        | 0x10002004 | 8           |
| TCNT | Watchdog Timer Counter           | RW | 0x????      | 0x10002008 | 16          |
| TCSR | Watchdog Timer Control Register  | RW | 0x0000      | 0x1000200C | 16          |

#### 22.2.1 Watchdog Control Register (TCSR)

The TCSR is a 16-bit read/write register. It contains the control bits for WDT. It is initialized to 0x00 by any reset.

| TCSR |          |    |    |    |    |    |   |   |   |   |          |   |        |        |        | 0x1000200C |  |  |  |
|------|----------|----|----|----|----|----|---|---|---|---|----------|---|--------|--------|--------|------------|--|--|--|
| Bit  | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5        | 4 | 3      | 2      | 1      | 0          |  |  |  |
|      | Reserved |    |    |    |    |    |   |   |   |   | PRESCALE |   | EXT_EN | RTC_EN | PCK_EN |            |  |  |  |
| RST  | 0        | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0        | 0 | 0      | 0      | 0      | 0          |  |  |  |

| Bits    | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW                       |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|------|-------|-------------|---|---|---|-----------------------|---|---|---|-----------------------|---|---|---|------------------------|---|---|---|------------------------|---|---|---|-------------------------|---|---|---|--------------------------|---------|--|--|----------|----|
| 15:6    | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | R                        |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 5:3     | PRESCALE | These bits select the TCNT count clock frequency.<br><table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Bit 2</th> <th>Bit1</th> <th>Bit 0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Internal clock: CLK/1</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Internal clock: CLK/4</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Internal clock: CLK/16</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Internal clock: CLK/64</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Internal clock: CLK/256</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Internal clock: CLK/1024</td> </tr> <tr> <td colspan="3">110~111</td> <td>Reserved</td> </tr> </tbody> </table> | Bit 2                    | Bit1 | Bit 0 | Description | 0 | 0 | 0 | Internal clock: CLK/1 | 0 | 0 | 1 | Internal clock: CLK/4 | 0 | 1 | 0 | Internal clock: CLK/16 | 0 | 1 | 1 | Internal clock: CLK/64 | 1 | 0 | 0 | Internal clock: CLK/256 | 1 | 0 | 1 | Internal clock: CLK/1024 | 110~111 |  |  | Reserved | RW |
| Bit 2   | Bit1     | Bit 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Description              |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 0       | 0        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Internal clock: CLK/1    |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 0       | 0        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Internal clock: CLK/4    |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 0       | 1        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Internal clock: CLK/16   |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 0       | 1        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Internal clock: CLK/64   |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 1       | 0        | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Internal clock: CLK/256  |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 1       | 0        | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Internal clock: CLK/1024 |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 110~111 |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reserved                 |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 2       | EXT_EN   | Select EXTAL as the timer clock input.<br>1: Enable<br>0: Disable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW                       |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 1       | RTC_EN   | Select RTCCLK as the timer clock input.<br>1: Enable<br>0: Disable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW                       |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |
| 0       | PCK_EN   | Select PCLK as the timer clock input.<br>1: Enable<br>0: Disable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW                       |      |       |             |   |   |   |                       |   |   |   |                       |   |   |   |                        |   |   |   |                        |   |   |   |                         |   |   |   |                          |         |  |  |          |    |

**NOTE:** The input clock of timer and the PCLK should keep to the rules as follows:

| Input clock of timer: IN_CLK                                  | Clock generated from the frequency divider (PRESCALE): DIV_CLK |
|---------------------------------------------------------------|----------------------------------------------------------------|
| PCK_EN == 0, RTC_EN == 1 and EXT_EN == 0<br>(IN_CLK = RTCCLK) | $f_{DIV\_CLK} < \frac{1}{2} f_{PCLK}$                          |
| PCK_EN == 0, RTC_EN == 0 and EXT_EN == 1<br>(IN_CLK = EXTAL)  | $f_{DIV\_CLK} < \frac{1}{2} f_{PCLK}$                          |
| PCK_EN == 1, RTC_EN == 0 and EXT_EN == 0<br>(IN_CLK = PCLK)   | ANY                                                            |

### 22.2.2 Watchdog Enable Register (TCER)

The TCER is an 8-bit read/write register. It contains the counter enable control bits for watchdog. It is initialized to 0x00 by any reset.

| TCER |  | 0x10002004 |  |  |  |  |  |   |   |          |   |   |   |   |   |      |
|------|--|------------|--|--|--|--|--|---|---|----------|---|---|---|---|---|------|
| Bit  |  |            |  |  |  |  |  | 7 | 6 | 5        | 4 | 3 | 2 | 1 | 0 |      |
|      |  |            |  |  |  |  |  |   |   | Reserved |   |   |   |   |   | TCEN |
| RST  |  |            |  |  |  |  |  | 0 | 0 | 0        | 0 | 0 | 0 | 0 | 0 |      |

| Bits | Name     | Description                                                  | RW |
|------|----------|--------------------------------------------------------------|----|
| 7:1  | Reserved | Writing has no effect, read as zero.                         | R  |
| 0    | TCEN     | Counter enable control.<br>0: Timer stop<br>1: Timer running | RW |

Note: writing TCEN to be zero can not stop the counter, you must stop it by writing TCU's register TSSR.WDTSS high. That is to say the counter can run when TCEN high and TCU's TSR low, but it can only stop by set TCU's TSR high.

### 22.2.3 Watchdog Timer Data Register (TDR)

The watchdog timer data register TDR is used to store the data to be compared with the content of the watchdog timer up-counter TCNT. This register can be directly read and written. (Default: indeterminate)

| TDR |     |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x10002000 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|-----|-----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| Bit |     |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
|     | TDR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| RST |     |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ?          | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |  |  |

### 22.2.4 Watchdog Timer Counter (TCNT)

The watchdog timer counter (TCNT) is a 16-bit read/write counter. The up-counter TCNT can be reset to 0 by software and counts up using the prescaler output clock. When TCNT count up to equal to TDR, the comparison match signal will be generated and a WDT reset is generated. The data can be read out at any time. The counter data can be written at any time. (Default: indeterminate)

| TCNT |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x10002008 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|------|------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| Bit  |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
|      | TCNT |  |  |  |  |  |  |  |  |  |  |  |  |  |  |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| RST  |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ?          | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |  |  |

## 22.3 Watchdog Timer Function

The following describes steps of using WDT:

- 1 Setting the PRESCALE of input clock in register TCSR.
- 2 Set register TDR and TCNT.
- 3 Select the input clock and enable the input clock in register TCSR.

After initialize the register of timer, we should start the counter as follows:

- 4 Set TCEN bit in TCER to 1. The counter TCNT begins to count.
- 5 If  $TCNT = TDR$ , a WDT reset will be generated.

**NOTES:**

- 1 The input clock and PCLK should follow the rules advanced before.
- 2 The clock of WDT can be stopped by setting register TSR, and register TSR can only be set by register TSSR or TSCR. The content of register TSR, TSSR and TSCR can be found in TCU spec.

## 23 PDMA Controller

Programmable DMA controller (PDMAC) is dedicated to smartly transfer data between on-chip peripherals (MSC, AIC, UART, etc.), external memories, and memory-mapped external devices.

### 23.1 Features

- Support up to 32 independent DMA channels
- Descriptor or No-Descriptor Transfer mode compatible with previous JZ SOC
- A simple Xburst-1 CPU supports smart transfer mode controlled by programmable firmware
- Transfer data units: 1-byte, 2-byte, 4-byte, 16-byte, 32-byte, 64-byte, 128-byte
- Transfer number of data unit:  $1 \sim 2^{24} - 1$
- Independent source and destination port width: 8-bit, 16-bit, 32-bit
- Fixed three priorities of channel groups: 0~3, highest; 4~11: mid; 12~31: lowest
- A dedicated bus interface - BIF interconnects with on-chip BCH
- A dedicated bus interface - NIF interconnects with on-chip NEMC or off-chip NEMC.
- An extra INTC IRQ can be bound to one programmable DMA channel
- Dedicated Security ROM and Security RAM supporting enhanced security requirements.

### 23.2 Block Diagram

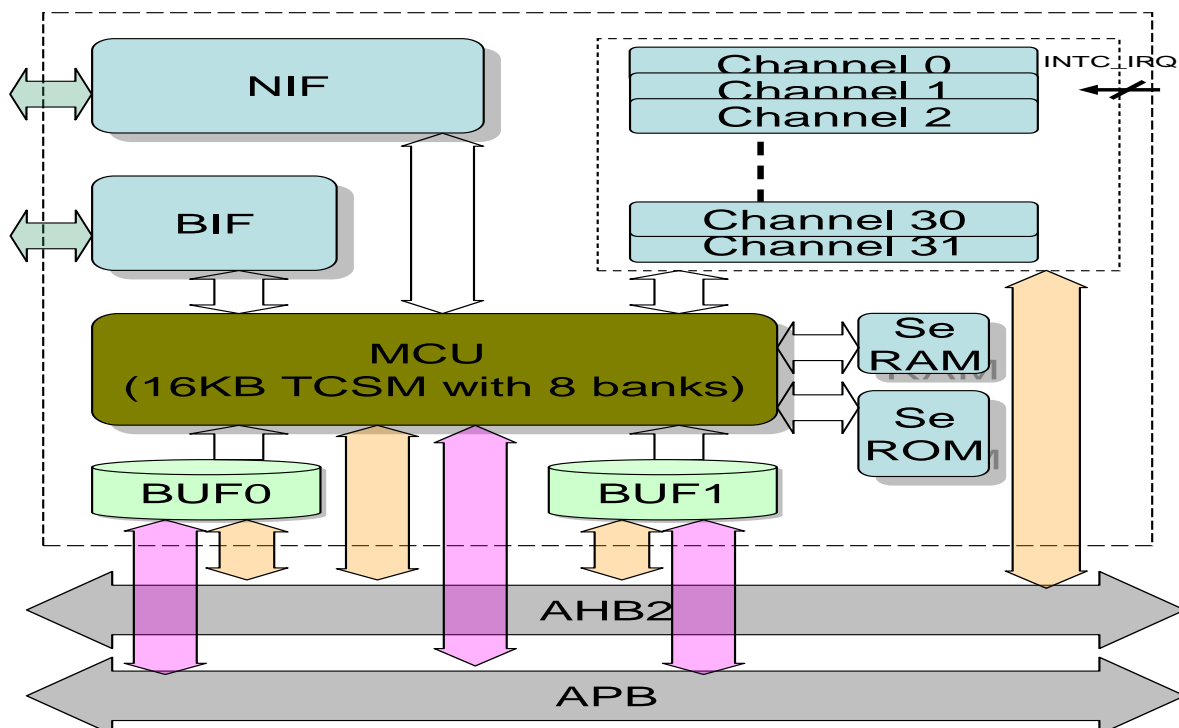


Figure 23-1 Block Diagram of PDMA

## 23.3 Memory Mapped Register Descriptions

### 23.3.1 DMA Channel Registers

Table 23-1 DMA Channel Registers (n=0~31)

| Name | Description                    | RW | Reset Value | Address             | Access Size (bit) |
|------|--------------------------------|----|-------------|---------------------|-------------------|
| DSAn | Channel n Source Address 0     | RW | 0x0         | 0x13420000 + n*0x20 | 32                |
| DTAn | Channel n Target Address 0     | RW | 0x0         | 0x13420004 + n*0x20 | 32                |
| DTCn | Channel n Transfer Count 0     | RW | 0x0         | 0x13420008 + n*0x20 | 32                |
| DRTn | Channel n Request Source 0     | RW | 0x0         | 0x1342000C + n*0x20 | 32                |
| DCSn | Channel n Control/Status 0     | RW | 0x0         | 0x13420010 + n*0x20 | 32                |
| DCMn | Channel n Command 0            | RW | 0x0         | 0x13420014 + n*0x20 | 32                |
| DDAn | Channel n Descriptor Address 0 | RW | 0x0         | 0x13420018 + n*0x20 | 32                |
| DSDn | Channel n Stride Difference 0  | RW | 0x0         | 0x1342001C + n*0x20 | 32                |

### 23.3.2 Global Control Registers

Table 23-2 Global Control Registers

| Name   | Description              | RW  | Reset Value | Address    | Access Size (bit) |
|--------|--------------------------|-----|-------------|------------|-------------------|
| DMAC   | DMA Control              | R/W | 0x0         | 0x13421000 | 32                |
| DIRQP  | DMA Interrupt            | R   | 0x0         | 0x13421004 | 32                |
| DDR    | DMA Doorbell             | R/W | 0x0         | 0x13421008 | 32                |
| DDRS   | DMA Doorbell Set         | W   | 0x0         | 0x1342100C | 32                |
| DCKE   | DMA Clock Enable         | W   | 0x0         | 0x13421010 | 32                |
| DCKES  | DMA Clock Enable Set     | W   | 0x0         | 0x13421014 | 32                |
| DCKEC  | DMA Clock Enable Clear   | W   | 0x0         | 0x13421018 | 32                |
| DMACP  | DMA Channel Programmable | R/W | 0x0         | 0x1342101C | 32                |
| DSIRQP | Channel soft IRQ to MCU  | R   | 0x0         | 0x13421020 | 32                |
| DSIRQM | Channel soft IRQ mask    | R   | 0xffffffff  | 0x13421024 | 32                |
| DCIRQP | Channel IRQ to MCU       | R   | 0x0         | 0x13421028 | 32                |
| DCIRQM | Channel IRQ to MCU mask  | R   | 0xffffffff  | 0x1342102C | 32                |
| DMCS   | MCU Control and Status   | RW  | 0xE1        | 0x13421030 | 32                |
| DMNMB  | MCU Normal MailBox       | RW  | ?           | 0x13421034 | 32                |
| DMSMB  | MCU Security MailBox     | RW  | ?           | 0x13421038 | 32                |
| DMINT  | MCU Interrupt            | RW  | 0x3         | 0x1342103C | 32                |

#### NOTES:

Grey ones are obsolete registers defined in previous JZ SOC. They are relative to clock gating and have no real function, and they are not supported any longer.

## 23.4 DMA Channel Register Definition

### 23.4.1 DMA Source Address (DSAn, n = 0 ~ 31)

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|            | <b>DSAn</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13420000 + 0x0 + (n*0x20)</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                                 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|            | SA          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b> | ?           | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                                  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description              | RW |
|------|------|--------------------------|----|
| 31:0 | SA   | Source physical address. | RW |

### 23.4.2 DMA Target Address (DTAn, n = 0 ~ 31)

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|            | <b>DTAn</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13420000 + 0x4 + (n*0x20)</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                                 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|            | TA          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b> | ?           | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                                  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description              | RW |
|------|------|--------------------------|----|
| 31:0 | TA   | Target physical address. | RW |

### 23.4.3 DMA Transfer Count (DTCn, n = 0 ~ 31)

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|            | <b>DTCn</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13420000 + 0x8 + (n*0x20)</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                                 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|            | Reserved    |    |    |    |    |    |    |    | TC |    |    |    |    |    |    |    |                                    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b> | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                                  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                      | RW |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:24 | Reserved | Write has no effect, read as zero.                                                                                                                                                                                                                                                               | R  |
| 23:0  | TC       | TC records the number of data unit to be transferred. Moreover, when Stride transfer mode is enabled, TC composes of two parts: the lower 16 bits is the number of data unit for sub-block transfer, the higher 8 bits is the number of sub-block. TC automatically counts down to 0 at the end. | RW |

### 23.4.4 DMA Request Types (DRTn, n = 0 ~ 31)

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|            | <b>DRTn</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13420000 + 0xC + (n*0x20)</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                                 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|            | Reserved    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RT                                 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b> | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                                  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | ? | ? | ? | ? | ? | ? |

| Bits | Name     | Description                        | RW |
|------|----------|------------------------------------|----|
| 31:6 | Reserved | Write has no effect, read as zero. | R  |
| 5:0  | RT       | Transfer request type.             | RW |

**Table 23-3 Transfer Request Types**

| RT5-0  | Description                                                                    |
|--------|--------------------------------------------------------------------------------|
| 000000 | Reserved.                                                                      |
| 000001 | Reserved.                                                                      |
| 000010 | I2S2 transmit-fifo-empty transfer request.                                     |
| 000011 | I2S2 receive-fifo-full transfer request.                                       |
| 000100 | I2S1 transmit-fifo-empty transfer request.                                     |
| 000101 | I2S1 receive-fifo-full transfer request.                                       |
| 000110 | I2S0 transmit-fifo-empty transfer request.                                     |
| 000111 | I2S0 receive-fifo-full transfer request.                                       |
| 001000 | Auto-request. (external address → external address)                            |
| 001001 | SADC receive-fifo-full transfer request                                        |
| 001010 | External request with DREQn. (external address ↔ external device with DACKn)   |
| 001011 | TCU channel n. (overflow interrupt, external address → external address space) |
| 001100 | UART4 transmit-fifo-empty transfer request. (external address → UTHR)          |
| 001101 | UART4 receive-fifo-full transfer request. (URBR → external address)            |
| 001110 | UART3 transmit-fifo-empty transfer request. (external address → UTHR)          |
| 001111 | UART3 receive-fifo-full transfer request. (URBR → external address)            |
| 010000 | UART2 transmit-fifo-empty transfer request. (external address → UTHR)          |
| 010001 | UART2 receive-fifo-full transfer request. (URBR → external address)            |
| 010010 | UART1 transmit-fifo-empty transfer request. (external address → UTHR)          |
| 010011 | UART1 receive-fifo-full transfer request. (URBR → external address)            |
| 010100 | UART0 transmit-fifo-empty transfer request. (external address → UTHR)          |
| 010101 | UART0 receive-fifo-full transfer request. (URBR → external address)            |
| 010110 | SSI0 transmit-fifo-empty transfer request.                                     |
| 010111 | SSI0 receive-fifo-full transfer request.                                       |
| 011000 | SSI1 transmit-fifo-empty transfer request.                                     |
| 011001 | SSI1 receive-fifo-full transfer request.                                       |



|                    |                                            |
|--------------------|--------------------------------------------|
| 011010             | MSC0 transmit-fifo-empty transfer request. |
| 011011             | MSC0 receive-fifo-full transfer request.   |
| 011100             | MSC1 transmit-fifo-empty transfer request. |
| 011101             | MSC1 receive-fifo-full transfer request.   |
| 011110             | MSC2 transmit-fifo-empty transfer request. |
| 011111             | MSC2 receive-fifo-full transfer request.   |
| 100000             | PCM0 transmit-fifo-empty transfer request. |
| 100001             | PCM0 receive-fifo-full transfer request.   |
| 100010             | PCM1 transmit-fifo-empty transfer request. |
| 100011             | PCM1 receive-fifo-full transfer request.   |
| 100100             | SMB0 transmit-fifo-empty transfer request. |
| 100101             | SMB0 receive-fifo-full transfer request.   |
| 100110             | SMB1 transmit-fifo-empty transfer request. |
| 100111             | SMB1 receive-fifo-full transfer request.   |
| 101000             | SMB2 transmit-fifo-empty transfer request. |
| 101001             | SMB2 receive-fifo-full transfer request.   |
| 101010             | SMB3 transmit-fifo-empty transfer request. |
| 101011             | SMB3 receive-fifo-full transfer request.   |
| 101100             | SMB4 transmit-fifo-empty transfer request. |
| 101101             | SMB4 receive-fifo-full transfer request.   |
| 101110             | DES transmit-fifo-empty transfer request.  |
| 101111             | DES receive-fifo-full transfer request.    |
| 110000 ~<br>111111 | reserved                                   |

**NOTES:**

- 1 Only auto request can be concurrently set in all channels with different source and target address.
- 2 For on-chip device DMA request, the corresponding source or target address that map to on-chip device must be set as fixed.

**23.4.5 DMA Channel Control/Status (DCSn, n = 0 ~ 31)**

| DCSn |      | 0x13420000 + 0x10 + (n*0x20) |     |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |    |   |   |    |     |     |     |   |   |
|------|------|------------------------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|----|---|---|----|-----|-----|-----|---|---|
| Bit  | 31   | 30                           | 29  | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8  | 7 | 6 | 5  | 4   | 3   | 2   | 1 | 0 |
|      | NDES | DES8                         | TOC |    |    |    |    |    |    |    |    |    |    |    |    |    | CDOA |    |    |    |    |    |   | AR |   |   | TT | HLT | TOE | CTE |   |   |
| RST  | ?    | ?                            | ?   | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?    | ?  | ?  | ?  | ?  | ?  | ? | ?  | 0 | 0 | 0  | 0   | 0   | 0   | 0 | 0 |

| Bits | Name | Description | RW |
|------|------|-------------|----|
|------|------|-------------|----|

|       |          |                                                                                                                                                                                                                                                                                                                                                                                                      |     |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 31    | NDES     | Descriptor or No-Descriptor Transfer Select.<br>0: Descriptor Transfer; 1: No-descriptor Transfer.                                                                                                                                                                                                                                                                                                   | RW  |
| 30    | DES8     | Descriptor 8 Word. 0: 4-word descriptor; 1: 8-word descriptor.                                                                                                                                                                                                                                                                                                                                       | RW  |
| 29:16 | TOC      | Time out counter. Available when TOE=1, undefined scratch field                                                                                                                                                                                                                                                                                                                                      | RW  |
| 15:8  | CDOA     | Copy of offset address of last completed descriptor from that in DMA command register. (The field is Ignored in No-Descriptor Transfer)                                                                                                                                                                                                                                                              | RW  |
| 7:5   | Reserved | Write has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                   | R   |
| 4     | AR       | Address Error. Hardware set it to 1 and software clear it to 0.<br>0: no address error; 1: address error occurred.                                                                                                                                                                                                                                                                                   | RW0 |
| 3     | TT       | Transfer Terminate.<br>0: Descriptor or No-Descriptor DMA transfer does not end<br>1: No-Link Descriptor or No-Descriptor DMA transfer end                                                                                                                                                                                                                                                           | RW  |
| 2     | HLT      | DMA halt. Hardware sets it to 1 when an abnormal case occurs during transfer, then software has to clear it to 0 for re-using the channel later.                                                                                                                                                                                                                                                     | RW0 |
| 1     | TOE      | Time out enable for transaction of a data unit.<br>0 – no time out monitor for a data unit’s transfer<br>1 – after a data unit has already been transferred but not completed yet, TOC field monitors the maximum cycles to be cost, and once exceeding the preset value occurs, a time-out error arises and HLT will be set 1 but TT keeps 0, moreover, if DCMn.TIE=1, a time-out interrupt arises. | RW  |
| 0     | CTE      | Channel transfer enable. 0: disable; 1: enable.                                                                                                                                                                                                                                                                                                                                                      | RW  |

**NOTES:**

- For a working channel, when an address error occurs or an on-going transfer is stopped by software writing 0 to CTE, HLT becomes 1 but TT keeps 0 that denotes an abnormal situation occurs.
- Software can stop any working channel at any time. However, after setting 0 to CTE, software must read TT, HLT and CTE to ensure the stop behavior has been done for the channel, that is,
  - TT=0, HLT=0, CTE =1, not halt yet, need polling again
  - TT=1, HLT=0, CTE=0, the channel just ends its work when software wants to stop it.
  - TT=0, HLT=1, CTE=0, the channel has been stopped but its transfer not complete yet.

**23.4.6 DMA Channel Command (DCMn, n = 0 ~ 31)**

|     | DCMn     |    |    |    |     |     |     |     |          |    |    |    |      |    |    |    | 0x13420000 + 0x14 + (n*0x20) |    |          |     |          |    |   |   |      |     |      |   |   |   |   |   |
|-----|----------|----|----|----|-----|-----|-----|-----|----------|----|----|----|------|----|----|----|------------------------------|----|----------|-----|----------|----|---|---|------|-----|------|---|---|---|---|---|
| Bit | 31       | 30 | 29 | 28 | 27  | 26  | 25  | 24  | 23       | 22 | 21 | 20 | 19   | 18 | 17 | 16 | 15                           | 14 | 13       | 12  | 11       | 10 | 9 | 8 | 7    | 6   | 5    | 4 | 3 | 2 | 1 | 0 |
|     | Reserved |    |    |    | SID | DID | SAI | DAI | Reserved |    |    |    | RDIL |    |    |    | SP                           | DP | Reserved | TSZ | Reserved |    |   |   | STDE | TIE | LINK |   |   |   |   |   |
| RST | 0        | 0  | 0  | 0  | 0   | 0   | 0   | 0   | ?        | ?  | 0  | 0  | ?    | ?  | ?  | ?  | ?                            | ?  | ?        | 0   | ?        | ?  | ? | 0 | 0    | 0   | 0    | 0 | 0 | 0 | 0 | 0 |

| Bits | Name | Description | RW |
|------|------|-------------|----|
|------|------|-------------|----|

|       |          |                                                                                                                                                                                                                                                  |    |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:28 | Reserved | Write has no effect, read as zero                                                                                                                                                                                                                | R  |
| 27:26 | SID      | Source identification. Only be used for special channel 0/1.<br>0: Source is TCSM.<br>1: Source is BCH (for channel 0) / NEMC (for channel 1)<br>2: Source is DDR (for channel 1) / Reserved (for channel 0)<br>3: Reserved.                     | RW |
| 25:24 | DID      | Destination identification. Only be used for special channel 0/1.<br>0: Destination is TCSM.<br>1: Destination is BCH (for channel 0) / NEMC (for channel 1)<br>2: Destination is DDR (for channel 1) / Reserved (for channel 0)<br>3: Reserved. | RW |
| 23    | SAI      | Source Address Increment. 0: no increment; 1: increment.                                                                                                                                                                                         | RW |
| 22    | DAI      | Destination Address Increment. 0: no increment; 1: increment.                                                                                                                                                                                    | RW |
| 19:16 | RDIL     | Recommended data unit size (unit: byte) for triggering device's DMA request when TSZ is autonomy.                                                                                                                                                | RW |
| 15:14 | SP       | Source port width. 00: 32-bit; 01: 8-bit; 10: 16-bit; 11: reserved.                                                                                                                                                                              | RW |
| 13:12 | DP       | Destination port width. 00: 32-bit; 01: 8-bit; 10: 16-bit; 11: reserved.                                                                                                                                                                         | RW |
| 11    | Reserved | Write has no effect, read as zero.                                                                                                                                                                                                               | R  |
| 10:8  | TSZ      | Transfer Data Size of a data unit. 000: 32-bit; 001: 8-bit; 010: 16-bit; 011: 16-byte; 100: 32-byte; 101: 64-byte; 110: 128-byte; 111: autonomy;                                                                                                 | RW |
| 7:3   | Reserved | Write has no effect, read as zero.                                                                                                                                                                                                               | R  |
| 2     | STDE     | Stride Disable/Enable. 0: address stride disable; 1: address stride enable.                                                                                                                                                                      | RW |
| 1     | TIE      | Transfer Interrupt Enable (TIE).<br>0: disable interrupt; 1: enable interrupt when TT is set to 1.                                                                                                                                               | RW |
| 0     | LINK     | Descriptor Link Enable. 0: disable; 1: enable.<br>(the field is ignored in No-Descriptor Transfer mode)                                                                                                                                          | RW |

**NOTES:**

SP and DP are only available for device FIFO or NEMC. But when transfer data between device and DDR memory, must set the port width of DDR side as the same as the device side; however, when both source and destination are in DDR memory, both SP and DP must be 32-bit by setting 0.

**Table 23-4 Available RDIL**

| RDIL | Description                       |
|------|-----------------------------------|
| 0    | Reserved                          |
| 1    | Recommended data unit is 1 bytes  |
| 2    | Recommended data unit is 2 bytes  |
| 3    | Recommended data unit is 3 bytes  |
| 4    | Recommended data unit is 4 bytes  |
| 5    | Recommended data unit is 8 bytes  |
| 6    | Recommended data unit is 16 bytes |
| 7    | Recommended data unit is 32 bytes |

|       |                                    |
|-------|------------------------------------|
| 8     | Recommended data unit is 64 bytes  |
| 9     | Recommended data unit is 128 bytes |
| 10~15 | Reserved                           |

**NOTES:**

1. if TSZ is autonomy, total bytes to be transferred equal the value of DCTn.TC (so maximum transfer bytes for autonomy are 16MB -1).
2. Programmer must care the detail of the FIFO of the device binding with relative DMA channel to set the correct and best recommended data unit value according to FIFO's data width and depth, refer to following table.

| FIFO form  | Best recommended RDIL                                |
|------------|------------------------------------------------------|
| 8x8bits    | 4 (half of total entries just accommodate 4 bytes)   |
| 8x16bits   | 5 (half of total entries just accommodate 8 bytes)   |
| 32x8bits   | 6 (half of total entries just accommodate 16 bytes)  |
| 16x16bits  | 6 (half of total entries just accommodate 16 bytes)  |
| 64x8bits   | 7 (half of total entries just accommodate 32 bytes)  |
| 64x32bits  | 9 (half of total entries just accommodate 128 bytes) |
| 128x16bits | 9 (half of total entries just accommodate 128 bytes) |
| 128x32bits | 9 (because maximum data unit is 128 bytes)           |

Moreover, programmer must be carefully obey following rules for correctness and best performance concern of DMA transfer:

1. **auto-request, both source and destination device are DDR memory.** No alignment constraint for address, however, both source port width and destination port width must be set 32bit, and transferred bytes in a bus transaction denoted by TSZ (when TSZ !=7) must not be less than 4. Setting autonomy (TSZ=7) is the best choice for transfer efficiency<sup>\*0</sup>
2. **auto-request, both source and destination device are NEMC,** then
  - a) if both source and destination port width are 8bit, no alignment constraint for address, setting autonomy (TSZ=7) is the best choice for transfer efficiency.
  - b) If either source port or destination port width is not 8bit, must not set autonomy to TSZ. The relative source or destination address must be 2-byte (port width is 16bit) or 4-byte (port width is 32bit) aligned. Moreover, the byte number denoted by TSZ must not be less than the least common multiple of the byte number denoted by source port width and destination port width. Larger TSZ means better transfer efficiency.
3. **auto-request, source device is NEMC while destination device is DDR memory,** then
  - a) if source port width is 8bit. no alignment constraint for address, setting autonomy (TSZ=7) is the best choice for transfer efficiency.
  - b) if source port width is 16bit, source address must be 2-byte aligned and TSZ must be the times of 2-byte (so must not set autonomy to TSZ); no alignment constraint for destination address, larger TSZ means better transfer efficiency.
  - c) If source port width is 32bit, source address must be 4-byte aligned and TSZ must be the times of 4-byte (so must not set autonomy to TSZ); no alignment constraint for destination

address, larger TSZ means better transfer efficiency.

4. **auto-request, source device is DDR memory while destination device is NEMC**, then
  - a) if destination port width is 8bit, no alignment constraint for address, setting autonomy (TSZ=7) is the best choice for transfer efficiency
  - b) if destination port width is 16bit, destination address must be 2-byte aligned and TSZ must be the times of 2-byte (so must not set autonomy to TSZ); no alignment constraint for source address, larger TSZ means better transfer efficiency.
  - c) If destination port width is 32bit, destination address must be 4-byte aligned and TSZ must be the times of 4-byte (so must not set autonomy to TSZ); no alignment constraint for source address, larger TSZ means better transfer efficiency.
5. **external device fifo receive request, destination device is DDR memory**, then no alignment constraint for destination address, and
  - a) if non-autonomy (TSZ !=7), for transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be times of 2-byte when source port width is 16bit or times of 4-byte when source port width is 32bit.
  - b) If autonomy (TSZ=7), for transferred bytes in a bus transaction denoted by RDIL, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be times of 2-byte when source port width is 16bit or times of 4-byte when source port width is 32bit.
6. **external device fifo receive request, destination device is NEMC**, then
  - a) if destination port width is 8bit, no alignment constraint for destination address, and if non-autonomy (TSZ !=7), for transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be times of 2-byte when source port width is 16bit or times of 4-byte when source port width is 32bit. If autonomy (TSZ=7), for transferred bytes in a bus transaction denoted by RDIL, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be times of 2-byte when source port width is 16bit or times of 4-byte when source port width is 32bit.
  - b) If destination port width is 16bit, must not set autonomy to TSZ and destination address must be 2-byte aligned. For transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must not be less than the least common multiple of the byte number denoted by source port width and destination port width.
  - c) If destination port width is 32bit, must not set autonomy to TSZ and destination address must be 4-byte aligned. For transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must not be less than the least common multiple of the byte number denoted by source port width and destination port width.
7. **external device fifo transmit request, source device is DDR memory**, then no alignment constraint for source address, and
  - a) if destination port width is 8bit, setting autonomy (TSZ=7) is the best choice for transfer efficiency
  - b) if destination port width is 16bit, when setting autonomy (TSZ=7), total transferred bytes

- must be the times of 2-byte, and for transferred bytes in a bus transaction denoted by RDIL, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be the times of 2-byte; when setting non-autonomy (TSZ!=7), for transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be the times of 2-byte.
- c) if destination port width is 32bit, when setting autonomy (TSZ=7), total transferred bytes must be the times of 4-byte, and for transferred bytes in a bus transaction denoted by RDIL, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be the times of 4-byte; when setting non-autonomy (TSZ!=7), for transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be the times of 4-byte.
8. **external device fifo transmit request, source device is NEMC**, then
- a) if source port width is 8bit, no alignment constraint for source address, and if non-autonomy (TSZ !=7), for transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be times of 2-byte when destination port width is 16bit or times of 4-byte when destination port width is 32bit.  
If autonomy (TSZ=7), for transferred bytes in a bus transaction denoted by RDIL, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must be times of 2-byte when destination port width is 16bit or times of 4-byte when destination port width is 32bit.
- b) If source port width is 16bit, must not set autonomy to TSZ and source address must be 2-byte aligned. For transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must not be less than the least common multiple of the byte number denoted by source port width and destination port width.
- c) If source port width is 32bit, must not set autonomy to TSZ and source address must be 4-byte aligned. For transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request<sup>\*1</sup> meanwhile must not be less than the least common multiple of the byte number denoted by source port width and destination port width.

**NOTES:**

<sup>\*0</sup> - when TSZ is not autonomy, total bytes to be transferred are bytes represented by TSZ \* TC, and each bus transaction can only transfer bytes represented by TSZ, so larger data unit is better for transfer efficiency. But since in practical, in most cases, total bytes to be transferred are not the times of the best data unit, or source and/or destination address violates alignment, thus **setting autonomy for TSZ is recommended** when condition is ok.

<sup>\*1</sup> - Please refer to relative device's spec for detail information of how to set critical trigger value.

### 23.4.7 DMA Descriptor Address (DDAn, n = 0 ~ 31)

This register is ignored in No-Descriptor Transfer mode.

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                     |    |    |    |    |    |          |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------------------------|----|----|----|----|----|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|
|            | <b>DDAn</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13420000 + 0x18 + (n*0x20)</b> |    |    |    |    |    |          |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                                  | 14 | 13 | 12 | 11 | 10 | 9        | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |   |   |
|            | DBA         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DOA                                 |    |    |    |    |    | Reserved |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b> | ?           | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                                   | ?  | ?  | ?  | ?  | ?  | ?        | ? | ? | ? | ? | ? | ? | ? | ? | ? | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                        | RW |
|-------|----------|------------------------------------|----|
| 31:12 | DBA      | Descriptor Base Address.           | RW |
| 11:4  | DOA      | Descriptor Offset Address.         | RW |
| 3:0   | Reserved | Write has no effect, read as zero. | R  |

### 23.4.8 DMA Stride Difference (DSDn, n = 0 ~ 31)

This register is ignored in No-Descriptor Transfer mode.

When address stride transfer mode is enabled in Descriptor mode, after a sub-block's transfer defined in DTCn completes, the source or target stride difference will be added to perform the new sub-block's start address, and the transfer will keep going until the transfer ends which means TC in DTCn reaches 0.

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|            | <b>DSDn</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13420000 + 0x1C + (n*0x20)</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                                  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |   |   |
|            | TSD         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SSD                                 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b> | ?           | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                                   | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits  | Name | Description                                                                                                                    | RW |
|-------|------|--------------------------------------------------------------------------------------------------------------------------------|----|
| 31:16 | TSD  | Target Stride Difference (next sub-block's start address – current sub-block's end address - 1), value range is -32768 ~ 32767 | RW |
| 15:0  | SSD  | Source Stride Difference (next sub-block's start address – current sub-block's end address - 1), value range is -32768 ~ 32767 | RW |

## 23.5 DMA Global Register Definition

### 23.5.1 DMA Control

| DMAC |      |      |       |       |      |          |    |    |    |    |       |    |    |    |    | 0x13421000 |          |    |    |    |    |    |   |   |   |   |     |    |      |      |   |   |   |
|------|------|------|-------|-------|------|----------|----|----|----|----|-------|----|----|----|----|------------|----------|----|----|----|----|----|---|---|---|---|-----|----|------|------|---|---|---|
| Bit  | 31   | 30   | 29    | 28    | 27   | 26       | 25 | 24 | 23 | 22 | 21    | 20 | 19 | 18 | 17 | 16         | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5   | 4  | 3    | 2    | 1 | 0 |   |
|      | FMSC | FSSI | FTSSI | FUART | FAIC | Reserved |    |    |    |    | INTCC |    |    |    |    | INTCE      | Reserved |    |    |    |    |    |   |   |   |   | HLT | AR | CH01 | DMAE |   |   |   |
| RST  | 0    | 0    | 0     | 0     | 0    | 0        | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0          | 0        | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0  | 0    | 0    | 0 | 0 | 0 |

| Bits  | Name    | Description                                                                                                                                                                                         | RW  |
|-------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 31    | FMSC    | 0: 0.5-data/clock for MSC on APB;1: 1-data/clock for MSC on APB.                                                                                                                                    | RW  |
| 30    | FSSI    | 0: 0.5-data/clock for SSI on APB;1: 1-data/clock for SSI on APB.                                                                                                                                    | RW  |
| 29    | FTSSI   | 0: 0.5-data/clock for TSSI on APB;1: 1-data/clock for TSSI on APB.                                                                                                                                  | RW  |
| 28    | FUART   | 0: 0.5-data/clock for UART on APB;1: 1-data/clock for UART on APB.                                                                                                                                  | RW  |
| 27    | FAIC    | 0: 0.5-data/clock for AIC on APB;1: 1-data/clock for AIC on APB.                                                                                                                                    | RW  |
| 21:17 | INTCC   | Which channel is bound with INTC_IRQ. Only available when INTCE is enabled                                                                                                                          | RW  |
| 16    | INTCE   | Permit INTC_IRQ to be bound to one of programmable channel.<br>0, INTC_IRQ is ignored; 1, INTC_IRQ is bound to channel n, n equals to INTCC.                                                        | RW  |
| 15:4  | Reserve | Write has no effect, read as zero.                                                                                                                                                                  | R   |
| 3     | HLT     | Global halt status, halt occurs in any channel, the bit should be set to 1 by hardware and cleared to 0 by software.<br>0: no halt occurred<br>1: halt occurred in one or more channels             | RW0 |
| 2     | AR      | Global address error status, address error occurs in any channel, the bit should be set to 1 by hardware and cleared to 0 by software.<br>0: no address error occurred<br>1: address error occurred | RW0 |
| 1     | CH01    | Special channel 0 and channel 1 enable. 1 –enable, 0 – disable.                                                                                                                                     | RW  |
| 0     | DMAE    | Global DMA enable.<br>0: disable DMA<br>1: enable DMA                                                                                                                                               | RW  |

**NOTES:**

- any of FMSC/FSSI/FTSSI/FUART/FAIC has been set, the corresponding DMA transfer on APB for MSC(MSC1, MSC2), SSI(SSI1), UART0~4, AIC will be in fast mode.
- DMAE is a global switch, so software must be careful to toggle it from 0 to 1 or 1 to 0. It is **UNPREDICTABLE** to set 0 to DMAE abruptly when some channels are still working.
- For more detail of CH01, refer to later chapter of special channel 0 and channel 1



### 23.5.2 DMA Interrupt Pending (DIRQP)

| DIRQP |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       | 0x13421004 |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit   | 31    | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16         | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|       | IRQ31 | IRQ30 | IRQ29 | IRQ28 | IRQ27 | IRQ26 | IRQ25 | IRQ24 | IRQ23 | IRQ22 | IRQ21 | IRQ20 | IRQ19 | IRQ18 | IRQ17 | IRQ16      | IRQ15 | IRQ14 | IRQ13 | IRQ12 | IRQ11 | IRQ10 | IRQ9 | IRQ8 | IRQ7 | IRQ6 | IRQ5 | IRQ4 | IRQ3 | IRQ2 | IRQ1 | IRQ0 |
| RST   | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                            | RW  |
|------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 31:0 | IRQn | IRQn denotes pending IRQ to main CPU for corresponding channel.<br>0: normal DMA transfer is in-progress or the channel is idle<br>1: an address error occurs (only available for UARTn) or normal DMA transfer done; moreover, if the relative channel is programmable, MCU can only set 1 to it.<br>Hardware can only set 1 to it and main CPU can only set 0 to it. | RW0 |

### 23.5.3 DMA Doorbell (DDB)

| DDB |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      | 0x13421008 |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |
|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit | 31   | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16         | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|     | DB31 | DB30 | DB29 | DB28 | DB27 | DB26 | DB25 | DB24 | DB23 | DB22 | DB21 | DB20 | DB19 | DB18 | DB17 | DB16       | DB15 | DB14 | DB13 | DB12 | DB11 | DB10 | DB9 | DB8 | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
| RST | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0          | 0    | 0    | 0    | 0    | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

| Bits | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW  |
|------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 31:0 | DBn  | DMA Doorbell. Software sets it to 1 and hardware clears it to 0.<br>0: disable DMA controller to fetch the first descriptor or DMA controller clears it to 0 as soon as it starts to fetch the descriptor<br>1: Writing 1 to DDS will set the corresponding DBn bit to 1 and enable DMA controller to fetch the first descriptor<br>For example, write 0x00000001 to DDS, DB0 bit is set to 1 and enable DMA channel 0 to fetch its first descriptor.<br>Write 0 to DDS, no meaning. | RW1 |

### 23.5.4 DMA Doorbell Set (DDS)

| DBS |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       | 0x1342100C |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit | 31    | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16         | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|     | DBS31 | DBS30 | DBS29 | DBS28 | DBS27 | DBS26 | DBS25 | DBS24 | DBS23 | DBS22 | DBS21 | DBS20 | DBS19 | DBS18 | DBS17 | DBS16      | DBS15 | DBS14 | DBS13 | DBS12 | DBS11 | DBS10 | DBS9 | DBS8 | DBS7 | DBS6 | DBS5 | DBS4 | DBS3 | DBS2 | DBS1 | DBS0 |
| RST | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

| Bits | Name | Description                                                                                                            | RW |
|------|------|------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | DBSn | DMA Doorbell Set for each channel. Read as zero, and when write:<br>0: ignore<br>1: Set the corresponding DBn bit to 1 | W1 |

### 23.5.5 DMA Channel Programmable (DMACP)

| DMACP |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       | 0x1342101C |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit   | 31    | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16         | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|       | DCP31 | DCP30 | DCP29 | DCP28 | DCP27 | DCP26 | DCP25 | DCP24 | DCP23 | DCP22 | DCP21 | DCP20 | DCP19 | DCP18 | DCP17 | DCP16      | DCP15 | DCP14 | DCP13 | DCP12 | DCP11 | DCP10 | DCP9 | DCP8 | DCP7 | DCP6 | DCP5 | DCP4 | DCP3 | DCP2 | DCP1 | DCP0 |
| RST   | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0          | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

| Bits | Name | Description                                                                                        | RW |
|------|------|----------------------------------------------------------------------------------------------------|----|
| 31:0 | DCPn | Channel programmable enable.<br>0, compatible with previous JZ SOC; 1, firmware controlled channel | RW |

### 23.5.6 DMA Soft IRQ Pending

| DSIRQP |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | 0x13421020 |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|------------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit    | 31     | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16         | 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|        | SIRQ31 | SIRQ30 | SIRQ29 | SIRQ28 | SIRQ27 | SIRQ26 | SIRQ25 | SIRQ24 | SIRQ23 | SIRQ22 | SIRQ21 | SIRQ20 | SIRQ19 | SIRQ18 | SIRQ17 | SIRQ16     | SIRQ15 | SIRQ14 | SIRQ13 | SIRQ12 | SIRQ11 | SIRQ10 | SIRQ9 | SIRQ8 | SIRQ7 | SIRQ6 | SIRQ5 | SIRQ4 | SIRQ3 | SIRQ2 | SIRQ1 | SIRQ0 |
| RST    | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0          | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

| Bits | Name  | Description                                                                                                                                                                                                  | RW |
|------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | SIRQn | Channel soft IRQ to MCU. Write is ignored<br>0, no soft IRQ to MCU; 1, pending soft IRQ to MCU.<br>Note that the register is read-only for other masters but is entirely controlled (R/W) by the MCU of PDMA | R  |

### 23.5.7 DMA Soft IRQ Mask

| DSIRQM |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | 0x13421024 |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|------------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit    | 31      | 30      | 29      | 28      | 27      | 26      | 25      | 24      | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16         | 15      | 14      | 13      | 12      | 11      | 10      | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|        | SIRQM31 | SIRQM30 | SIRQM29 | SIRQM28 | SIRQM27 | SIRQM26 | SIRQM25 | SIRQM24 | SIRQM23 | SIRQM22 | SIRQM21 | SIRQM20 | SIRQM19 | SIRQM18 | SIRQM17 | SIRQM16    | SIRQM15 | SIRQM14 | SIRQM13 | SIRQM12 | SIRQM11 | SIRQM10 | SIRQM9 | SIRQM8 | SIRQM7 | SIRQM6 | SIRQM5 | SIRQM4 | SIRQM3 | SIRQM2 | SIRQM1 | SIRQM0 |
| RST    | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1          | 1       | 1       | 1       | 1       | 1       | 1       | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      |

| Bits | Name   | Description                                                                                                                                                                                                                                    | RW |
|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | SIRQMn | Channel soft IRQ mask. Write is ignored<br>0, not mask corresponding channel's soft IRQ; 1, mask corresponding channel's soft IRQ<br>Note that the register is read-only for other masters but is entirely controlled (R/W) by the MCU of PDMA | R  |

### 23.5.8 DMA Channel IRQ Pending to MCU

| DCIRQP |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | 0x13421028 |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|------------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit    | 31     | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16         | 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|        | CIRQ31 | CIRQ30 | CIRQ29 | CIRQ28 | CIRQ27 | CIRQ26 | CIRQ25 | CIRQ24 | CIRQ23 | CIRQ22 | CIRQ21 | CIRQ20 | CIRQ19 | CIRQ18 | CIRQ17 | CIRQ16     | CIRQ15 | CIRQ14 | CIRQ13 | CIRQ12 | CIRQ11 | CIRQ10 | CIRQ9 | CIRQ8 | CIRQ7 | CIRQ6 | CIRQ5 | CIRQ4 | CIRQ3 | CIRQ2 | CIRQ1 | CIRQ0 |
| RST    | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0          | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

| Bits | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW |
|------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | CIRQn | Channel IRQ Pending to MCU. The register is read-only<br>0, no channel IRQ to MCU; 1, pending channel IRQ to MCU<br>When channel n is not programmable, the CIRQn is ignored by hardware. When channel n is programmable and DCMn.TIE is active 1, DIRQP.IRQn can not be set automatically any more after address error occurs or TT becomes active 1, instead, if DCIRQMn==0, CIRQn may be set 1 immediately to raise an IRQ to MCU. Moreover, when an channel m with above attribute is bound with the INTC_IRQ, if DCIRQMm==0, an active INTC_IRQ request will trigger an active CIRQn immediately | R  |

### 23.5.9 DMA Channel IRQ to MCU Mask

|            |               |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |                   |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
|------------|---------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|-------------------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|            | <b>DCIRQM</b> |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | <b>0x1342102C</b> |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
| <b>Bit</b> | 31            | 30      | 29      | 28      | 27      | 26      | 25      | 24      | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16      | 15                | 14      | 13      | 12      | 11      | 10      | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|            | CIRQM31       | CIRQM30 | CIRQM29 | CIRQM28 | CIRQM27 | CIRQM26 | CIRQM25 | CIRQM24 | CIRQM23 | CIRQM22 | CIRQM21 | CIRQM20 | CIRQM19 | CIRQM18 | CIRQM17 | CIRQM16 | CIRQM15           | CIRQM14 | CIRQM13 | CIRQM12 | CIRQM11 | CIRQM10 | CIRQM9 | CIRQM8 | CIRQM7 | CIRQM6 | CIRQM5 | CIRQM4 | CIRQM3 | CIRQM2 | CIRQM1 | CIRQM0 |
| <b>RST</b> | 1             | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1                 | 1       | 1       | 1       | 1       | 1       | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      |

| Bits | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                        | RW |
|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | CIRQMn | Mask of Channel IRQ to MCU. The register is read-only for other master but is entirely controlled (R/W) by the MCU of PDMA<br>0, not mask corresponding channel's IRQ to MCU; 1, mask corresponding channel's IRQ to MCU<br>When channel n is not programmable, the CIRQMn is ignored by hardware.<br>When channel n is programmable and DCMn.TIE is active 1, CIRQMn can mask corresponding channel's IRQ to MCU. | R  |

### 23.5.10 Programmable Channel Bound With INTC\_IRQ

In this SOC, all peripheral IRQs originally being issued to main CPU now can be taken over by PDMA through binding special INTC\_IRQ to one of programmable channel. Note that INTC\_IRQ from INTC may be different to normal IRQ of INTC to main CPU, please refer to the spec of INTC for detail of how to configure INTC\_IRQ sources. In practical, maybe it is a better tradeoff to let MCU of PDMA take over trivial transactions of some devices from main CPU.

### 23.5.11 Special Channel 0 and Channel 1

When a product integrating this SOC need use raw NAND chipset properly through PDMA, the only method is setting special channel 0 and 1, following key bit fields must be preset:

1. DMAC.CH01 must be set 1
2. DCM0.TIE and DCM1.TIE must be set 1
3. DCM0.LINK and DCM1.LINK must be set 0
4. DCS0.NDES and DCS1.NDES must be set 1
5. DCM0.STDE and DCM1.STDE must be set 0
6. DCM0.SAI and DCM0.DAI must be set 1
7. DCM1.SAI and DCM1.DAI must be set 1
8. DCM0.SP and DCM0.DP must be set 0
9. DCM1.SP and DCM1.DP must be set 0
10. DCM0.TSZ and DCM1.TSZ must be set 7
11. DMACP.DCP0 and DMACP.DCP1 must be set 1 after relative firmware has been prepared.

## 23.6 MCU

MCU in PDMA is a mini CPU compatible with Xburst-1 ISA without implementing CACHE, MMU, DEBUG, FPU and MXU. It is very similar as the AUX in VPU but has different memory mapped control and status register.

### 23.6.1 MCU Control & Status

| DMCS |       |      |          |    |    |    |    |    |        |    |    |    |    |    |    | 0x13421030 |    |    |     |    |    |         |         |   |   |   |        |   |   |   |   |   |
|------|-------|------|----------|----|----|----|----|----|--------|----|----|----|----|----|----|------------|----|----|-----|----|----|---------|---------|---|---|---|--------|---|---|---|---|---|
| Bit  | 31    | 30   | 29       | 28 | 27 | 26 | 25 | 24 | 23     | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13  | 12 | 11 | 10      | 9       | 8 | 7 | 6 | 5      | 4 | 3 | 2 | 1 | 0 |
|      | SLEEP | SCMD | Reserved |    |    |    |    |    | SC_OFF |    |    |    |    |    |    |            |    |    | BCH |    |    | BTB_INV | SC_CALL |   |   |   | SW_RST |   |   |   |   |   |
| RST  | 0     | 0    | 0        | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0   | 0  | 0  | 0       | 0       | 0 | 1 | 1 | 1      | 0 | 0 | 0 | 0 | 1 |

| Bits  | Name     | Description                                                                                                                                      | RW |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | SLEEP    | Sleep status of MCU. Read only by main CPU.                                                                                                      | R  |
| 30    | SCMD     | Security Mode. Read only by main CPU.                                                                                                            | R  |
| 29:24 | reserved | Write has no effect, read as zero.                                                                                                               | R  |
| 23:8  | SC_OFF   | Set the offset of the caller's data structure for SC_CALL                                                                                        | RW |
| 7     | BCH_DB   | Block syndrome of BCH decoding, low level active                                                                                                 | R  |
| 6     | BCH_DF   | BCH decoding finished, low level active                                                                                                          | R  |
| 5     | BCH_EF   | BCH encoding finished, low level active                                                                                                          | R  |
| 4     | BTB_INV  | Writing 1 invalidates BTB in MCU. Writing 0 has no effect, read as zero.                                                                         | W  |
| 3     | SC_CALL  | SecurityCall. Writing 1 triggers a security exception which directs MCU to run security code located in the on-chip security ROM., read as zero. | W1 |
| 2:1   | reserved | Write has no effect, read as zero.                                                                                                               | R  |
| 0     | SW_RST   | Software reset.<br>1, MCU keeps at reset state; 0, do not soft-reset MCU any more                                                                | RW |

#### NOTES:

BCH\_DB, BCH\_DF and BCH\_EF are only available when special channel 0 and channel 1 is used for implementing storage of raw NAND chipsets. Please refer to BCH spec for detail.

### 23.6.2 MCU Normal MailBox

| DMNMB |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x13421034 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | MB |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST   | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?          | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description                                                                                                                                              | RW |
|------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | MB   | Contents of Normal MailBox. Writing any value to the register will trigger a normal mailbox IRQ to main CPU when relative MASK field in DMINT is not set | RW |

### 23.6.3 MCU Security MailBox

|              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| <b>DMSMB</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x13421038</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>   | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|              | MB |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>   | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                 | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description                                                                                                                                                                                                                             | RW |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | MB   | Contents of Security MailBox. Writing any value to the the register will trigger a security mailbox IRQ to main CPU when relative MASK field in DMINT is not set. Note that writing the register is inhibited by HW when DMCS.SCMD == 0 | RW |

### 23.6.4 MCU Interrupt

|              |          |    |    |    |    |    |    |    |    |    |    |    |    |    |      |                   |          |    |    |    |    |    |   |   |   |   |   |   |   |   |        |        |   |   |
|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|-------------------|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|--------|--------|---|---|
| <b>DMINT</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |    |      | <b>0x1342103C</b> |          |    |    |    |    |    |   |   |   |   |   |   |   |   |        |        |   |   |
| <b>Bit</b>   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17   | 16                | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1      | 0      |   |   |
|              | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | S_IP | N_IP              | Reserved |    |    |    |    |    |   |   |   |   |   |   |   |   | S_IMSK | N_IMSK |   |   |
| <b>RST</b>   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0                 | 0        | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0      | 0      | 1 | 1 |

| Bits  | Name     | Description                                                                                                                           | RW  |
|-------|----------|---------------------------------------------------------------------------------------------------------------------------------------|-----|
| 31:18 | Reserved | Write has no effect, read as zero.                                                                                                    | R   |
| 17    | S_IP     | Security Mailbox IRQ pending. 1-pending IRQ. Hardware only set 1 by writing DMSMB when DMCS.SCMD == 1. Can be clear to 0 in any mode. | RW0 |
| 16    | N_IP     | Normal Mailbox IRQ pending. 1-pending IRQ. Hardware only set 1 by writing DMNMB. Can be clear to 0 in any mode.                       | RW0 |
| 15:2  | Reserved | Write has no effect, read as zero.                                                                                                    | R   |
| 1     | S_IMSK   | Security Mailbox IRQ mask. 1-mask; 0-not mask<br>Can only be modified in security mode.                                               | RW  |
| 0     | N_IMSK   | Normal Mailbox IRQ mask. 1-mask; 0-not mask                                                                                           | RW  |

### 23.6.5 Multiple Bank Tightly Coupled Sharing Memory

In MCU, there is a Tightly Coupled Sharing Memory (TCSM) with 16KB size and total 8 banks. Multiple- bank TCSM permits several masters accessing different banks in parallel. TCSM works at the same speed as MCU and both instruction fetch and data read/write to TCSM by MCU can be performed in one cycle.

**Table 23-5 TCSM space**

|       | VA (only available for MCU) | PA (not remapped)     | PA (remapped)         |
|-------|-----------------------------|-----------------------|-----------------------|
| BANK0 | F4000000h ~ F4007FFh        | 13422000h ~ 134227FFh | F3422000h ~ F34227FFh |
| BANK1 | F4000800h ~ F400FFFh        | 13422800h ~ 13422FFFh | F3422800h ~ F3422FFFh |
| BANK2 | F4001000h ~ F40017FFh       | 13423000h ~ 134237FFh | F3423000h ~ F34237FFh |
| BANK3 | F4001800h ~ F4001FFFh       | 13423800h ~ 13423FFFh | F3423800h ~ F3423FFFh |
| BANK4 | F4002000h ~ F40027FFh       | 13424000h ~ 134247FFh | F3424000h ~ F34247FFh |
| BANK5 | F4002800h ~ F4002FFFh       | 13424800h ~ 13424FFFh | F3424800h ~ F3424FFFh |
| BANK6 | F4003000h ~ F40037FFh       | 13425000h ~ 134257FFh | F3425000h ~ F34257FFh |
| BANK7 | F4003800h ~ F4003FFFh       | 13425800h ~ 13425FFFh | F3425800h ~ F3425FFFh |

### 23.6.6 Security ROM & Security RAM

In the PDMA, there is a 32KB on-chip security ROM with start programming address at 0xF5000000 and an 4KB on-chip security RAM with start programming address at 0xF5010000. Both the ROM and the RAM are only accessible in SMD mode by MCU.

### 23.6.7 CP0 Registers of MCU

Available CP0 registers of MCU are not entirely compatible with MIPS PRA spec, they are listed below.

| CP0 Name (number) | Description                                                                                                                                                                                          | slt |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| STATUS (12)       | Bit 31~3: reserved, read as zero<br>Bit2: ERL, Bit1: EXL, Bit0: IE                                                                                                                                   | 0   |
| INTCTL (12)       | Bit31~3: reserved, read as zero;<br>Bit2: IRQ pending of soft IRQ to MCU;<br>Bit1: IRQ pending of channel IRQ to MCU;<br>Bit0: IRQ pending of INTC_IRQ<br>Note that the register is read-only by MCU | 1   |
| CAUSE (13)        | Bit31:BD, whether the interrupted PC is located in branch delay slot<br>Bit30 ~ 0: reserved, read as zero                                                                                            | 0   |
| SeCAUSE (13)      | Bit31:BD, similar as CAUSE, but being active when an IRQ interrupted a running SecurityCall routine.<br>Bit30 ~ 0: reserved, read as zero                                                            | 1   |
| EPC (14)          | IRQ Exception program counter                                                                                                                                                                        | 0   |

|                     |                                    |   |
|---------------------|------------------------------------|---|
| <i>SePC (24)</i>    | Security Exception program counter | 0 |
| <i>ErrorPC (30)</i> | Reset exception program counter    | 0 |

### 23.6.8 Normal Exceptions Accepted by MCU

There are two kinds of normal exceptions that can be accepted by MCU: RESET and IRQ.

Note that RESET exception has the handler entry: 0xF4000000, while IRQ has another one: 0xF4000100.

Once the MCU accepts a normal exception, according to the definition of XBurst-1 Programming Manual, current interrupted PC will be pushed into CP0.EPC or CP0.ErrPC, and further CP0.STATUS.ERL or CP0.SATUS.EXL will be automatically set 1. Firmware can take use of MTC0/MFC0 to access these three CP0 registers: EPC, ErrPC and STATUS. Firmware must use ERET instruction to return to the interrupted locale to resume executing.

### 23.6.9 How to Boot MCU Up

After power-on reset, MCU keeps at reset state due to DMCS.SW\_RST==1. To boot MCU up, first of all, RESET exception handler must be loaded into TCSM at the physical address range beginning from 0x13422000 or 0xF3422000 for remapped case. Programmer can use main CPU or some Non-programmable DMA channel to finish this loading work. Then clear DMCS.SW\_RST to 0 by main CPU so that MCU can acknowledge a RESET exception, now MCU begins to run from the PC 0xF4000000. Following is a simple example.

1. Prepare a simple RESET exception handler, total 3 instructions:
 

```

1: WAIT //sleep
B 1b //endless loop
NOP //delay slot

```
  2. Load the handler into TCSM at the physical address range beginning from 0x13422000 or 0xF3422000 for remapped case.
  3. Clear DMCS.SW\_RST to 0 by main CPU
- Now MCU will execute WAIT instruction and then enter sleep state

### 23.6.10 SecurityCall Accepted by MCU

SecurityCall is an especial exception that can be acknowledged by MCU. There are only two modes for MCU: normal mode and security mode that can be identified by DMCS.SCMD. For convenience of description, simplified term **NMD** (normal mode) and **SMD** (security mode) will be used in later statements. When MCU is running in NMD mode (DMCS.SCMD == 0), a SecurityCall exception can be granted immediately when both CP0.STATUS.ERL and CP0.SATUS.EXL are zero. In detail, do as following steps:

1. load SecurityCall relative data structure to the proper location of TCSM, must be 256-byte aligned
2. set DMCS.SC\_OFF to point to the position (0xF4000000 + DMCS.SC\_OFF<<8).
3. write 1 to DMCS.SC\_CALL to try to trigger a SecurityCall exception.



4. When DMCS.SCMD == 1, which means SecurityCall is successfully acknowledged, the handler entry is 0xF5000000.

Note that active DMCS.SCMD always inhibit new attempt of SecurityCall, thus, the service routine for SecurityCall is not re-enterable, therefore OS running in main CPU must carefully manage the SecurityCall through such as semaphore mechanism.

### 23.6.11 Interruptable SMD Mode

When SecurityCall is acknowledged, CP0.SATUS.EXL also will be automatically set 1 to inhibit MCU responding any active IRQ to MCU meanwhile CP0.EPC will record the interrupted PC located in TCSM. However, since some functions of SecurityCall may spend long time, so code running in SMD mode should permit being interrupted by IRQ coming occasionally. Therefore, some service routines in Security ROM should clear CP0.SATUS.EXL to 0 if necessary, and after that, once an IRQ is acknowledged in SMD mode, the corresponding handler entry becomes 0xF5000100 other than 0xF4000100, meanwhile an SW invisible flag SC\_IF will be automatically set 1 and a special CP0 register CP0.SC\_EPC (only accessible by MFC0/MTC0 in SMD mode) will record the interrupted PC located in Security ROM. The code sequence located at 0XF5000100 should be elaborately constructed like following:

```

1. subu sp, sp, 4*32 //prepare to push GPR
2. mfc0 k0, $14, 0 //EPC (return PC in NMD mode) to k0
3. mfc0 k1, $24, 0 //SePC (return PC in SMD mode) to k1
4. sw ra, 4*31(sp) //push ra
5. jal _push_gpr //subroutine of pushing GPR (except ra)
6. nop
7. la k0, _se_ram_top //top of Security RAM
8. sw sp, 0(k0) //sp in SMD mode is always saved at top of Security RAM
9. la k0, RET_POS //exit after IRQ in SMD mode to k0
10. la k1, 0xF4000100 //IRQ entry in NMD mode to k1
11. mtc0 k1, $14, 0
12. mtc0 k0, $24, 0
13. MACRO_OF_CHAOS_GPR
14. la sp, _se_ram_top
15. subu sp, sp, 4*32
16. lw sp, 4*26(sp) //get sp in NMD mode
17. eret //*1
RET_POS: //*2
18. la k0, _se_ram_top
19. jal _pop_gpr //subroutine of popping GPR (except ra)
20. lw sp, 0(k0) //restore sp in SMD mode
21. lw ra, 4*31(sp) //pop ra
22. addu sp, sp, 4*32
23. mtc0 k0, $14, 0

```

```

24. mtc0 k1, $24, 0
25. mfc0 k0, $12, 0
26. mtc0 k1, $30, 0 //SePC to ErrPC
27. ori k0, k0, 4
28. mtc0 k0, $12, 0 //set ERL to Status
29. ERET //3

```

**NOTES:**

**\*1** After the execution of ERET, MCU enters NMD mode, SC\_IF retains 1 and CP0.STATUS.EXL retains 1

**\*2** When SC\_IF is active 1 and MCU is in NMD mode, executing ERET always restores SMD mode and returns to the PC pointed by CP0.SC\_EPC, furthermore, HW will automatically set 1 to CP0.STATUS.EXL and set 0 to CP0.STATUS.ERL and set 0 to SC\_IF. Note that SC\_IF being 1 always inhibits any new attempt of SecurityCall.

**\*3** When CP0.STATUS.ERL is 1, ERET does not clear SMD mode. Therefore, for code in Security ROM, DO NOT set 1 to CP0.STATUS.ERL through MTC0 instruction except here.

**23.7 DMA manipulation****23.7.1 Descriptor Transfer Mode****23.7.1.1 Non-Stride (1-Dimensional) Transfer Mode**

To do proper Descriptor DMA transfer in a channel, do following steps:

- 1 Check whether the status of DMA controller is available, that is, for global control (DMAC), ensure that DMAC.AR=0 and DMAC.HLT=0; and then for expected channel n, ensure that DCSn.AR=0, DCSn.HLT=0, DCSn.TT=0, DTCn=0.
- 2 guarantee DCSn.NDES=0, and select 4 word or 8 word descriptor by DCSn.DES8.
- 3 Build descriptor in memory. Write the first descriptor's address in DDAn and the address must be 16Bytes aligned for 4word descriptor and 32Bytes aligned for 8word descriptor, otherwise, the hardware behavior later is **UNPREDICTABLE**. The address includes two parts: Base and Offset address. If the descriptor is linked, the 32-bit address of next descriptor is composed of 20-bit Base address in DDAn and 8-bit Offset address in DESn.CDOA. See Table 23-6 Descriptor Structure for the detailed descriptor structure.
- 4 Don't update DRTn manually by software, instead, hardware will manage the request type of descriptor's transfer automatically.
- 5 Set 1 to the corresponding bit in DDB to initiate descriptor fetch.
- 6 Set DMAC.DMAE=1 and expected DCSn.CTE=1 to activate channel n.
- 7 Hardware will automatically clear the corresponding bit in DDB as soon as it starts to fetch the descriptor.
- 8 Wait for DMA request from peripherals to start channel n's DMA transfer.
- 9 After DMA transfers described by the current descriptor complete, if current DCMn.LINK =0, then DCSn.TT will be set to 1 immediately. Further, if DCMn.TIE=1, it will issue an interrupt

request. However, if DCMn.LINK=1, next descriptor will be automatically fetched to repeat step 8 and 9 but ignore setting DCSn.TT until a final descriptor with setting of DCMn.LINK=0 has been done.

- 10 When channel n's transfer ends normally (DCSn.TT=1) or abnormally (DCSn.HLT=1), to correctly reuse the channel later, software must close the channel first by setting 0 to DCSn.CTE, and then safely clear DCSn.TT or DCSn.HLT to 0.

**Table 23-6 Descriptor Structure**

| Word       | Bit   | Name                   | Function                                  |
|------------|-------|------------------------|-------------------------------------------|
| 1st (DES0) | 31    | EACKS                  | External DMA DACKn output polarity select |
|            | 30    | EACKM                  | External DMA DACKn output Mode select     |
|            | 29-28 | ERDM                   | External DMA request detection Mode       |
|            | 27-24 | Reserved               | --                                        |
|            | 23    | SAI                    | Source Address Increment                  |
|            | 22    | DAI                    | Target Address Increment                  |
|            | 21-20 | Reserved               | --                                        |
|            | 19-16 | RDIL                   | Request Detection Interval Length         |
|            | 15-14 | SP                     | Source port width                         |
|            | 13-12 | DP                     | Target port width                         |
|            | 11    | Reserved               | --                                        |
|            | 10-8  | TSZ                    | Transfer Data Size                        |
|            | 7-3   | Reserved               | --                                        |
|            | 2     | STDE                   | Stride transfer enable                    |
|            | 1     | TIE                    | Transfer Interrupt Enable                 |
| 0          | LINK  | Descriptor Link Enable |                                           |
| 2nd (DES1) | 31-0  | DSA                    | Source Address                            |
| 3rd (DES2) | 31-0  | DTA                    | Target Address                            |
| 4th (DES3) | 31-24 | DOA                    | Descriptor Offset address                 |
|            | 23-0  | DTC                    | Transfer Counter                          |
| 5th (DES4) | 31-16 | TSD                    | Target Stride Address                     |
|            | 15-0  | SSD                    | Source Stride Address                     |
| 6th (DES5) | 31-6  | Reserved               | --                                        |
|            | 5-0   | DRT                    | DMA Request Type                          |
| 7th (DES6) | 31-0  | Reserved               | --                                        |
| 8th (DES7) | 31-0  | Reserved               | --                                        |

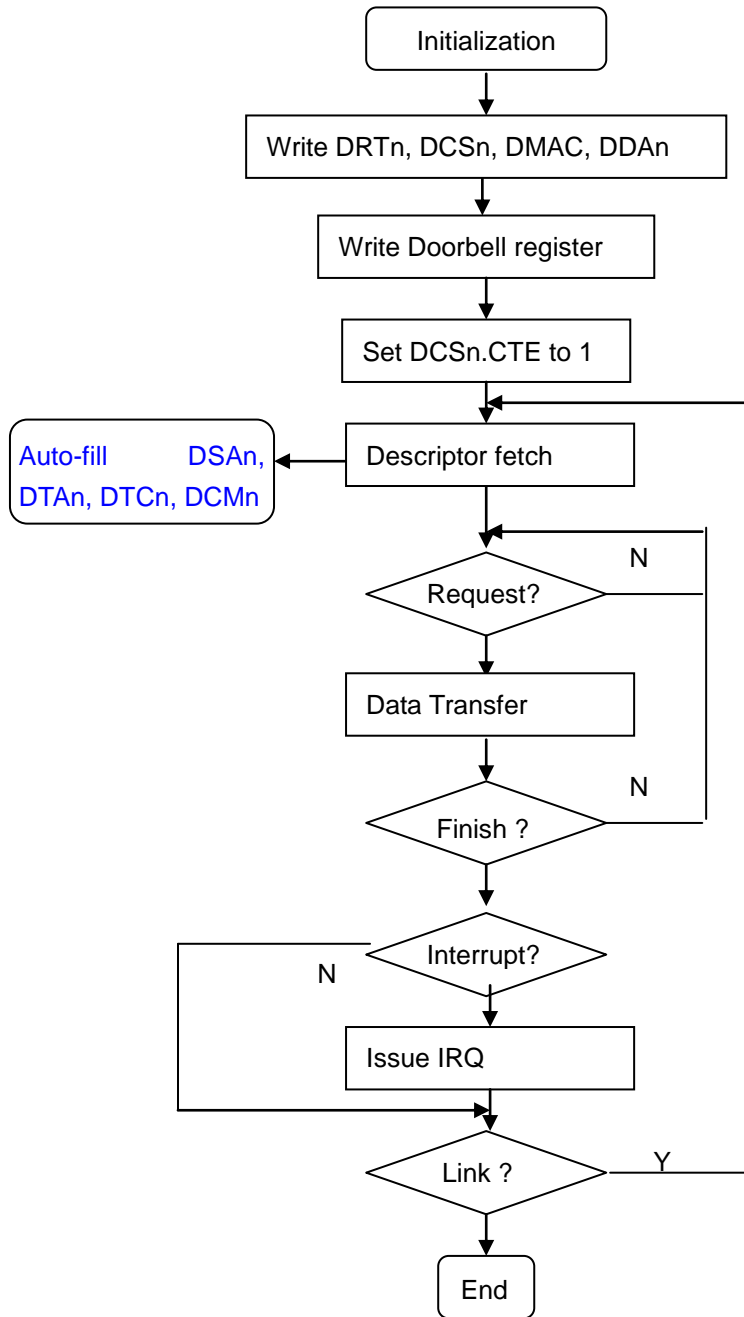
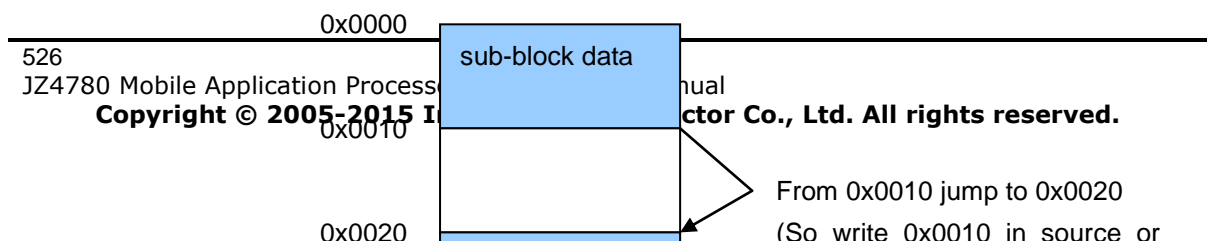


Figure 23-2 Descriptor Transfer Flow

### 23.7.1.2 Stride (2-dimensional) Transfer Mode

Stride transfer mode means total transfer counts are subdivided into dozens of or even hundreds of sub-blocks with same size, and each adjacent two sub-blocks has fixed nonzero address gap between the tail of preceding one and the head of subsequent one.



### Figure 23-3 Example for Stride Transfer Mode

Manipulation steps of Stride Transfer Mode are similar to Non-stride Transfer Mode. However, each new sub-block's start address (source or destination) is performed by adding the next adjacent byte address of very completed sub-block's end address to corresponding stride difference.

#### 23.7.2 No-Descriptor Transfer Mode

To do proper No-Descriptor DMA transfer, do following steps:

- 1 Check whether the status of DMA controller is available, that is, for global control (DMAC), ensure that DMAC.AR=0 and DMAC.HLT=0; and then for expected channel n, ensure that DCSn.AR=0, DCSn.HLT=0, DCSn.TT=0, DTCn=0.
- 2 Forx channel n, initialize DSA<sub>n</sub>, DTAn, DTCn, DRTn, DCSn, DCMn properly.
- 3 Set DMAC.DMAE=1 and DCSn.CTE=1 to launch DMA transfer.

For a channel with auto-request (DRTn.RT=0x8), the transfer begins automatically when the DCSn.CTE bit and DMAC.DMAE bit are set to 1. While for a channel with other request types, the transfer does not start until a transfer request is issued and detected.

For any channel n, The DTCn value is decremented by 1 for each successful transaction of a data unit. When the specified number of transfer data unit has been completed (DTCn = 0), the transfer ends normally. Meanwhile if DCMn.TIE=1, corresponding bit of DIRQP will be set to 1 to raise an interrupt request. However, during the transfer, if a DMA address error occurs, the transfer should be suspended, both DCSn.AR and DMAC.AR are set to 1 as well as corresponding bit of DIRQP to raise an interrupt request despite of DCMn.TIE.

Sometimes, for example, an UART parity error occurs for a channel that is transferring data between such UART and another terminal. In the case, both DCSn.HLT and DMAC.HLT are set to 1 and the transfer is suspended. Software should identify halt status by checking such two bits and re-configure DMA to let DMA rerun properly later.

## 23.8 DMA Requests

DMA transfer requests are normally generated from either the data transfer source or target, but also they can be issued by on-chip peripherals that are neither the source nor the target. There are

two DMA transfer request types: auto-request, and on-chip peripheral request. For any channel n, its transfer request type is determined through DRTn.

### 23.8.1 Auto Request

When there is no explicit transfer request signal available, for example, memory-to-memory transfer or memory to some on-chip peripherals like GPIO, the auto-request mode allows the DMA to automatically generate a transfer request signal internally. Therefore, when DMA initialization done, once the DMAC.DMAE and DCSn.CTE are set to 1, the transfer begins immediately in channel n which DRTn=0x8.

### 23.8.2 On-Chip Peripheral Request

In the mode, transfer request signals come from on-chip peripherals or even off-chip devices. When a device issues a peripheral request to its corresponding channel n, the transferred byte number for the request is equals to:

1. byte number determined by the bytes denoted by DCMn.TSZ when DCMn.TSZ != 7
2. byte number determined by the bytes denoted by RDIL when DCMn.TSZ == 7

Be careful that the above number must not exceed the device's expected one otherwise FIFO under-run or over-run may occur. However, if the number is too small than the expected one, DMA's bus efficiency will become worse.

## 23.9 How to Use Programmable DMA Channel

When system needs to setup one programmable channel for later use of smart data transfer, do following steps:

1. Load elaborate firmware to TCSM, should use physical address 0x13422xxx. Common code fragment of IRQ exception handler must be located at physical address range beginning from 0x13422100.
2. Prepare a DMA channel n using expected transfer mode as description in preceding chapters
3. Must set DCMn.TIE=1 and DMACP.DCPn=1
4. Must ensure DCIRQM.CIRQMn==0
5. Launch the channel n as description in preceding chapters

Now the channel n becomes a working programmable channel. It is the responsibility of local channel IRQ's handler to remove corresponding channel IRQ source by setting 0 to DCSn.TT.

Moreover, if programmer wants to let the channel can be triggered by local soft IRQ, DSIRQM.SIRQMn must be set 0, then at expected moment, the MCU sets 1 to DSIRQP.SIRQn to trigger a soft IRQ. It is the responsibility of this soft IRQ's handler to remove the IRQ by setting 0 to DSIRQP.SIRQn.

## 24 SAR A/D Controller

### 24.1 Overview

The A/D is CMOS low-power dissipation 12bit touch screen SAR analog to digital converter. It operates with 3.3/1.1V power supply. It is developed as an embedded high resolution ADC targeting to the 40nm CMOS process and has wide application in portable electronic devices, high-end home entertainment center, communication systems and so on.

The SAR A/D controller is dedicated to control A/D to work at three different modes: Touch Screen (measure pen position and pen down pressure), Battery (check the battery power), and two auxiliary input. Touch Screen can transfer the data to memory though the DMA or CPU. Battery and two auxiliary input can transfer the data to memory though CPU.

Features:

- 7 Channels
- Resolution: 12-bit
- Integral nonlinearity:  $\pm 1$  LSB
- Differential nonlinearity:  $\pm 0.5$  LSB
- Resolution/speed: up to 2Msps
- Max Frequency: 200k
- Low power dissipation: 1.5mW(worst)
- Support 4-wire and 5-wire touch panel measurement (Through pin XP, XN, YP, YN and AUX2)
- Support multi-touch detect
- Support write control command by software
- Support voltage measurement (Through pin VBAT)
- Support two auxiliary input (Through pin AUX1, AUX2)
- Single-end and Differential Conversion Mode
- Auto X/Y, X/Y/Z1/Z2 and X/Y/Z1/Z2/X2/Y2 position measurement
- Support external touch screen controller
- Pin Description

**Table 24-1 SADC Pin Description**

| Name | I/O | Description                                        |
|------|-----|----------------------------------------------------|
| XN   | AI  | Touch screen analog differential X- position input |
| YN   | AI  | Touch screen analog differential Y- position input |
| XP   | AI  | Touch screen analog differential X+ position input |
| YP   | AI  | Touch screen analog differential Y+ position input |
| VBAT | AI  | VBAT direct input <sup>*1</sup>                    |
| AUX1 | AI  | Auxiliary Input                                    |
| AUX2 | AI  | Auxiliary Input                                    |

**NOTE:**

\*<sup>1</sup>: Users who already deployed resistor networks on board level can use VBAT to directly measure the battery value.

## 24.2 Register Description

In this section, we will describe the registers in SAR A/D controller. Following table lists all the register definitions. All registers' 32bit addresses are physical addresses. And detailed function of each register will be described below.

**Table 24-2 SADC Register Description**

| Name    | Description                    | RW | Reset Value | Address    | Access Size |
|---------|--------------------------------|----|-------------|------------|-------------|
| ADENA   | ADC Enable Register            | RW | 0x80        | 0x10070000 | 8           |
| ADCFG   | ADC Configure Register         | RW | 0x00040000  | 0x10070004 | 32          |
| ADCTRL  | ADC Control Register           | RW | 0x3F        | 0x10070008 | 8           |
| ADSTATE | ADC Status Register            | RW | 0x00        | 0x1007000C | 8           |
| ADSAME  | ADC Same Point Time Register   | RW | 0x0000      | 0x10070010 | 16          |
| ADWAIT  | ADC Wait Time Register         | RW | 0x0000      | 0x10070014 | 16          |
| ADTCH   | ADC Touch Screen Data Register | RW | 0x00000000  | 0x10070018 | 32          |
| ADV DAT | ADC VBAT Data Register         | RW | 0x0000      | 0x1007001C | 16          |
| ADADAT  | ADC AUX Data Register          | RW | 0x0000      | 0x10070020 | 16          |
| ADCLK   | ADC Clock Divide Register      | RW | 0x00041000  | 0x10070028 | 32          |
| ADCMD   | ADC Command Register           | RW | 0x00000000  | 0x10070024 | 32          |

### 24.2.1 ADC Enable Register (ADENA)

The register ADENA is used to trigger A/D to work.

| ADENA |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10070000 |       |        |          |        |       |        |       |   |   |   |   |   |   |   |   |   |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|-------|--------|----------|--------|-------|--------|-------|---|---|---|---|---|---|---|---|---|
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15    | 14     | 13       | 12     | 11    | 10     | 9     | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | POWER | SLP_MD | Reserved | PENDEN | TCHEN | VBATEN | AUXEN |   |   |   |   |   |   |   |   |   |
| RST   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0     | 0      | 0        | 0      | 0     | 0      | 0     | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                       | RW |
|------|----------|-------------------------------------------------------------------|----|
| 31:8 | Reserved | Writing has no effect, read as zero.                              | R  |
| 7    | POWER    | SADC Power control bit.<br>1: SADC power down<br>0: SADC power on | RW |



|     |          |                                                                                                                                                                                                       |    |
|-----|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|     |          | When POWER is set from 1 to 0, you should wait at least 2ms to enable SADC.                                                                                                                           |    |
| 6   | SLP_MD   | SLEEP Mode Control.<br>1:Enter sleep mode<br>0:Exit sleep mode                                                                                                                                        | RW |
| 5:4 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                  | R  |
| 3   | PENDEN   | Pen Down Detect Enable control.<br>0: disable<br>1: enable                                                                                                                                            | RW |
| 2   | TCHEN    | Touch Screen Enable Control.<br>0: disable<br>1: enable                                                                                                                                               | RW |
| 1   | VBATEN   | VBAT Enable Control.<br>No matter TCHEN is 1 or 0, VBATEN can be set to 1 to sample the voltage of battery, and when the value of voltage is ready, PBATEN will be cleared by hardware auto.          | RW |
| 0   | AUXEN    | AUX n Enable Control.<br>No matter TCHEN is 1 or 0, AUXEN can be set to 1 to sample the voltage of AUX1, AU2 or AUX3, and when the value of voltage is ready. AUXEN will be cleared by hardware auto. | RW |

**NOTES:**

- 1 TCHEN, VBATEN and AUXEN can be set to 1 at the same time. The priority of the three mode is AUX > VBAT > TCH.
- 2 SLP\_MD, TCHEN can be set to 1 at the same time. The priority of the two mode is SLP\_MD > TCH.
- 3 When VBATEN and AUXEN are all 0, SLP\_MD can be set to 1.

**24.2.2 ADC Configure Register (ADCFG)**

The register ADCFG is used to configure the A/D.

| ADCFG |      |          |    |    |    |    |    |    |            |           |     |    |    |    |        | 0x10007004 |      |          |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
|-------|------|----------|----|----|----|----|----|----|------------|-----------|-----|----|----|----|--------|------------|------|----------|----|----|----|----|---|---|-----|---|---|---|---|---|---|---|
| Bit   | 31   | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23         | 22        | 21  | 20 | 19 | 18 | 17     | 16         | 15   | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | SPZZ | Reserved |    |    |    |    |    |    | Wire_sel_r | Cmd_sel_r | PRU |    |    |    | DMA_EN | XYZ        | SNUM | Reserved |    |    |    |    |   |   | CMD |   |   |   |   |   |   |   |
| RST   | 0    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0         | 0   | 0  | 0  | 0  | 1      | 0          | 0    | 0        | 0  | 0  | 0  | 0  | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name               | Description                                                                                                                                                                                                                             | RW |
|------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31   | SPZZ <sup>*1</sup> | The X <sub>d</sub> Y <sub>d</sub> Z <sub>m</sub> Z <sub>n</sub> of different point measure can be different.<br>But the X <sub>d</sub> Y <sub>d</sub> Z <sub>m</sub> Z <sub>n</sub> of the same point measure can be same or different. | RW |

|       |                                                                                                                                                                                   | <p>0: The <math>X_d Y_d Z_m Z_n</math> of the same point measure is all the same<br/>(<math>X_d Y_d Z1Z2, X_d Y_d Z1Z2, X_d Y_d Z1Z2, X_d Y_d Z1Z2 \dots X_d Y_d Z1Z2</math>)</p> <p>1: The <math>X_d Y_d Z_m Z_n</math> of the same point measure maybe different<br/>(<math>X_d Y_d Z1Z2, X_d Y_d Z3Z4, X_d Y_d Z3Z4, X_d Y_d Z1Z2 \dots X_d Y_d Z1Z2</math>)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|----------|-----|-----------------------|-----|-----------------------|-----|--------------------------------------------------------------------------------------------------------------------|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|---|-----|---|-----|----------|-----|----------|----|
| 30:24 | Reserved                                                                                                                                                                          | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | R    |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 23    | WIRE_SEL                                                                                                                                                                          | <p>0: use 4-wire touch panel</p> <p>1: use 5-wire touch panel</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RW   |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 22    | CMD_SEL                                                                                                                                                                           | <p>0: use hardware inter command to control touch panel</p> <p>1: use software write command to control touch panel</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW   |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 21:16 | RPU                                                                                                                                                                               | <p>Internal Pull-up resistor for Pen Detection.</p> <p>6'b111111: 64kΩ/63 = 1.02kΩ (least sensitive)</p> <p>6'b111110: 64KΩ/62 = 1.03KΩ</p> <p>... (pull-up = 64kΩ / binary value of RPU)</p> <p>6'b000010: 64KΩ/2 = 32KΩ</p> <p>6'b000001: 64kΩ/1 = 64kΩ (most sensitive) default</p> <p>6'b000000: RESERVED (do not use this setting)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW   |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 15    | DMA_EN                                                                                                                                                                            | <p>When A/D is used as Touch Screen , DMA_EN is used as follows:</p> <p>0: The sample data is read by CPU</p> <p>1: The sample data is read by DMA</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW   |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 14:13 | XYZ                                                                                                                                                                               | <p>When A/D is used in Touch Screen mode, XYZ is used as follows:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">XYZ</th> <th>Measure</th> </tr> </thead> <tbody> <tr> <td>00</td> <td><math>X_s \rightarrow Y_s</math></td> </tr> <tr> <td>01</td> <td><math>X_d \rightarrow Y_d</math></td> </tr> <tr> <td>10</td> <td><math>X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d</math> or <math>X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d</math></td> </tr> <tr> <td>11</td> <td><math>X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d \rightarrow X2 \rightarrow Y2</math> or<br/><math>X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d \rightarrow X2 \rightarrow Y2</math></td> </tr> </tbody> </table> | XYZ  | Measure  | 00  | $X_s \rightarrow Y_s$ | 01  | $X_d \rightarrow Y_d$ | 10  | $X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d$ or $X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d$ | 11  | $X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d \rightarrow X2 \rightarrow Y2$ or<br>$X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d \rightarrow X2 \rightarrow Y2$ | RW  |   |     |   |     |          |     |          |    |
| XYZ   | Measure                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 00    | $X_s \rightarrow Y_s$                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 01    | $X_d \rightarrow Y_d$                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 10    | $X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d$ or $X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d$                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 11    | $X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d \rightarrow X2 \rightarrow Y2$ or<br>$X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d \rightarrow X2 \rightarrow Y2$ |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 12:10 | SNUM                                                                                                                                                                              | <p>The number of repeated sampling one point. When A/D is used as Touch Screen, SNUM is used as follows:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">SNUM</th> <th>Number</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Reserved</td> </tr> <tr> <td>001</td> <td>1</td> </tr> <tr> <td>010</td> <td>2</td> </tr> <tr> <td>011</td> <td>3</td> </tr> <tr> <td>100</td> <td>4</td> </tr> <tr> <td>101</td> <td>5</td> </tr> <tr> <td>110</td> <td>Reserved</td> </tr> <tr> <td>111</td> <td>Reserved</td> </tr> </tbody> </table>                                                                                                                                                                                                 | SNUM | Number   | 000 | Reserved              | 001 | 1                     | 010 | 2                                                                                                                  | 011 | 3                                                                                                                                                                                 | 100 | 4 | 101 | 5 | 110 | Reserved | 111 | Reserved | RW |
| SNUM  | Number                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 000   | Reserved                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 001   | 1                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 010   | 2                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 011   | 3                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 100   | 4                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 101   | 5                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 110   | Reserved                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 111   | Reserved                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 9:2   | Reserved                                                                                                                                                                          | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | R    |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 1:0   | CMD                                                                                                                                                                               | <p>CMD is used to choose the current sample command when ADENA.AUXEN is set to 1.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 40%;">CMD</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>Reserved</td> </tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | CMD  | Function | 00  | Reserved              | RW  |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| CMD   | Function                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |
| 00    | Reserved                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |          |     |                       |     |                       |     |                                                                                                                    |     |                                                                                                                                                                                   |     |   |     |   |     |          |     |          |    |

|  |  |    |                      |
|--|--|----|----------------------|
|  |  | 01 | Measure AUX1 voltage |
|  |  | 10 | Measure AUX2 voltage |
|  |  | 11 | Reserved             |

**NOTE:**

\*1: X<sub>s</sub>, Y<sub>s</sub> means the reference mode of X, Y is single-end mode.

X<sub>d</sub>, Y<sub>d</sub>, Z1<sub>d</sub>, Z2<sub>d</sub>, Z3<sub>d</sub>, Z4<sub>d</sub> means the reference mode of X, Y, Z1, Z2, Z3, Z4 is differential mode.

When you measure X<sub>s</sub> you need to make sure that X-plate is driven by external DC power.

When you measure Y<sub>s</sub> you need to make sure that Y-plate is driven by external DC power.

### 24.2.3 ADC Control Register (ADCTRL)

The register ADCTRL is used to control A/D to work.

| ADCTRL |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10070008 |    |    |    |    |    |    |   |   |   |         |       |       |       |       |       |   |   |
|--------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---------|-------|-------|-------|-------|-------|---|---|
| Bit    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6       | 5     | 4     | 3     | 2     | 1     | 0 |   |
|        | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   | SLPENDM | PENDM | PENUM | DTCHM | VRDYM | ARDYM |   |   |
| RST    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0       | 0     | 1     | 1     | 1     | 1     | 1 | 1 |

| Bits | Name     | Description                                                        | RW |
|------|----------|--------------------------------------------------------------------|----|
| 31:6 | Reserved | Writing has no effect, read as zero.                               | R  |
| 5    | SLPENDM  | In SLEEP mode pen down interrupt mask.<br>0: enabled<br>1: masked  | RW |
| 4    | PENDM    | Pen down interrupt mask.<br>0: enabled<br>1: masked                | RW |
| 3    | PENUM    | Pen up interrupt mask.<br>0: enabled<br>1: masked                  | RW |
| 2    | DTCHM    | Touch Screen Data Ready interrupt mask.<br>0: enabled<br>1: masked | RW |
| 1    | VRDYM    | VBAT data ready interrupt mask.<br>0: enabled<br>1: masked         | RW |
| 0    | ARDYM    | AUX data ready interrupt mask.<br>0: enabled<br>1: masked          | RW |

### 24.2.4 ADC Status Register (ADSTATE)

The register ADSTATE is used to keep the status of A/D.

| ADSTATE |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x1007000C |         |          |        |      |      |      |      |      |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |
|---------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|---------|----------|--------|------|------|------|------|------|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|--|
| Bit     | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15      | 14       | 13     | 12   | 11   | 10   | 9    | 8    | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |  |  |  |  |  |  |  |  |  |  |  |
|         | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | SLP_RDY | Reserved | SLPEND | PEND | PENU | DTCH | VRDY | ARDY |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |
| RST     | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0       | 0        | 0      | 0    | 0    | 0    | 0    | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |

| Bits | Name     | Description                                                                                                             | RW |
|------|----------|-------------------------------------------------------------------------------------------------------------------------|----|
| 31:8 | Reserved | Writing has no effect, read as zero.                                                                                    | R  |
| 7    | SLP_RDY  | Sleep state bit.<br>1:The set of sleep mode is ready<br>0:The set of sleep mode is not ready                            | R  |
| 6    | Reserved | Writing has no effect, read as zero.                                                                                    | R  |
| 5    | SLPEND   | In SLEEP mode pen down interrupt flag. Write 1 to this bit, the bit will clear this bit.<br>1: active<br>0: not active  | RW |
| 4    | PEND     | Pen down interrupt flag. Write 1 to this bit, the bit will clear this bit.<br>1: active<br>0: not active                | RW |
| 3    | PENU     | Pen up interrupt flag. Write 1 to this bit, the bit will clear this bit.<br>1: active<br>0: not active                  | RW |
| 2    | DTCH     | Touch screen data ready interrupt flag. Write 1 to this bit, the bit will clear this bit.<br>1: active<br>0: not active | RW |
| 1    | VRDY     | VBAT data ready interrupt flag. Write 1 to this bit, the bit will clear this bit.<br>1: active<br>0: not active         | RW |
| 0    | ARDY     | AUX data ready interrupt flag. Write 1 to this bit, the bit will clear this bit.<br>1: active<br>0: not active          | RW |

### 24.2.5 ADC Same Point Time Register (ADSAME)

The register ADSAME is used to store the interval time between repeated sampling the same point.

The clock of the counter is clk\_us.

| ADSAME |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10070010 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SCNT       |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 24.2.6 ADC Wait Pen Down Time Register (ADWAIT)

The register ADWAIT is used to store the interval time of wait pen down. And the register can be used as the interval time among the different point. The clock of the counter is clk\_ms.

| ADWAIT |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10070014 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    | WCNT       |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |

### 24.2.7 ADC Touch Screen Data Register (ADTCH)

The read-only ADTCH is corresponded to 16x32 bit FIFO, it keep the sample data for touch screen. 0~11 bits are data, 15 bit is data type. 16~27 bits are data, 31 bit is data type. When write to the register, DATA will be clear to 0.

| ADTCH |       |          |    |        |    |    |    |    |    |    |    |    |    |    |    | 0x10070018 |          |    |        |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|-------|----------|----|--------|----|----|----|----|----|----|----|----|----|----|----|------------|----------|----|--------|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31    | 30       | 29 | 28     | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15       | 14 | 13     | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | TYPE1 | Reserved |    | TDATA1 |    |    |    |    |    |    |    |    |    |    |    | TYPE0      | Reserved |    | TDATA0 |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST   | 0     | 0        | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0        | 0  | 0      | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |

| Bits | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW |
|------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31   | TYPE1 | Type of the Touch Screen Data1.<br>When A/D is used as Touch Screen, ADCFG.XYZ=10 or XYZ=11:<br>TYPE1=1: $X_d \rightarrow Y_d \rightarrow Z1 \rightarrow Z2$ or $X_d \rightarrow Y_d \rightarrow Z1 \rightarrow Z2 \rightarrow X2 \rightarrow Y2$ ;<br>TYPE1=0: $X_d \rightarrow Y_d \rightarrow Z3 \rightarrow Z4$ or $X_d \rightarrow Y_d \rightarrow Z3 \rightarrow Z4 \rightarrow X2 \rightarrow Y2$ .<br>When A/D is used as Touch Screen, ADCFG.XYZ=00 or XYZ=01: | RW |

|       |          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|       |          | TYPE1=0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |    |
| 30:28 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                | R  |
| 27:16 | TDATA1   | The concert data of touch screen A/D.                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW |
| 15    | TYPE0    | Type of the Touch Screen Data2.<br>When A/D is used as Touch Screen, ADCFG.XYZ=10 or XYZ=11:<br>TYPE0=1: $X_d \rightarrow Y_d \rightarrow Z1 \rightarrow Z2$ or $X_d \rightarrow Y_d \rightarrow Z1 \rightarrow Z2 \rightarrow X2 \rightarrow Y2$ ;<br>TYPE0=0: $X_d \rightarrow Y_d \rightarrow Z3 \rightarrow Z4$ or $X_d \rightarrow Y_d \rightarrow Z3 \rightarrow Z4 \rightarrow X2 \rightarrow Y2$ .<br>When A/D is used as Touch Screen, ADCFG.XYZ=00 or XYZ=01:<br>TYPE0=0. | RW |
| 14:12 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                | R  |
| 11:0  | TDATA0   | The concert data of touch screen A/D.                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW |

**NOTES:**

- 1 When A/D is used as Touch Screen, ADCFG.XYZ=00.  
The format of touch screen data is as follows:

| Type1 | Reserved | Data1 | Type0 | Reserved | Data0 |
|-------|----------|-------|-------|----------|-------|
| 0     | 000      | $Y_s$ | 0     | 000      | $X_s$ |

- 2 When A/D is used as Touch Screen, ADCFG.XYZ=01.  
The format of touch screen data is as follows:

| Type1 | Reserved | Data1 | Type0 | Reserved | Data0 |
|-------|----------|-------|-------|----------|-------|
| 0     | 000      | $Y_d$ | 0     | 000      | $X_d$ |

- 3 When A/D is used as Touch Screen, ADCFG.XYZ=10.TYPE=1.  
The format of touch screen data is as follows:

| Type1 | Reserved | Data1  | Type0 | Reserved | Data0  |
|-------|----------|--------|-------|----------|--------|
| 1     | 000      | $Y_d$  | 1     | 000      | $X_d$  |
| 1     | 000      | $Z2_d$ | 1     | 000      | $Z1_d$ |

Users need to read twice to get the whole data. The first time reading gets the data  $Y_d$  and  $X_d$ . The second time reading gets the data  $Z2_d$  and  $Z1_d$ .

The touch pressure measurement formula is as follows: (You can use formula 1 or formula 2.)

$$R_{TOUCH} = R_{X-Plate} \cdot \frac{X-Position}{4096} \left( \frac{Z_2}{Z_1} - 1 \right) \quad (1)^{*1}$$

$$R_{TOUCH} = \frac{R_{X-Plate} \cdot X-Position}{4096} \left( \frac{4096}{Z_1} - 1 \right) - R_{Y-Plate} \cdot \left( 1 - \frac{Y-Position}{4096} \right) \quad (2)^{*1}$$

- 4 When A/D is used as Touch Screen, ADCFG.XYZ=10.TYPE=0.

The format of touch screen data is as follows:

| Type1 | Reserved | Data1  | Type0 | Reserved | Data0  |
|-------|----------|--------|-------|----------|--------|
| 0     | 000      | $Y_d$  | 0     | 000      | $X_d$  |
| 0     | 000      | $Z4_d$ | 0     | 000      | $Z3_d$ |

Users need to read twice to get the whole data. The first time reading gets the data  $Y_d$  and  $X_d$ . The second time reading gets the data  $Z4_d$  and  $Z3_d$ .

The touch pressure measurement formula is as follows: (You can use formula 3 or formula 4.)

$$R_{TOUCH} = R_{Y-Plate} \cdot \frac{Y-Position}{4096} \left( \frac{Z_4}{Z_3} - 1 \right) \quad (3)^{*1}$$

$$R_{TOUCH} = \frac{R_{Y-Plate} \cdot Y-Position}{4096} \left( \frac{4096}{Z_3} - 1 \right) - R_{X-Plate} \cdot \left( 1 - \frac{X-Position}{4096} \right) \quad (4)^{*1}$$

- 5 When A/D is used as Touch Screen, ADCFG.XYZ=11.TYPE=1.

The format of touch screen data is as follows:

| Type1 | Reserved | Data1  | Type0 | Reserved | Data0  |
|-------|----------|--------|-------|----------|--------|
| 1     | 000      | $Y_d$  | 1     | 000      | $X_d$  |
| 1     | 000      | $Z2_d$ | 1     | 000      | $Z1_d$ |
| 1     | 000      | $Y2$   | 1     | 000      | $X2$   |

Users need to read thrice to get the whole data. The first time reading gets the data  $Y_d$  and  $X_d$ . The second time reading gets the data  $Z2_d$  and  $Z1_d$ . The third time reading gets the data  $Y2$  and  $X2$ .

- 6 When A/D is used as Touch Screen, ADCFG.XYZ=11.TYPE=0.

The format of touch screen data is as follows:

| Type1 | Reserved | Data1  | Type0 | Reserved | Data0  |
|-------|----------|--------|-------|----------|--------|
| 0     | 000      | $Y_d$  | 0     | 000      | $X_d$  |
| 0     | 000      | $Z4_d$ | 0     | 000      | $Z3_d$ |
| 0     | 000      | $Y2$   | 0     | 000      | $X2$   |

Users need to read thrice to get the whole data. The first time reading gets the data  $Y_d$  and  $X_d$ . The second time reading gets the data  $Z4_d$  and  $Z3_d$ . The third time reading gets the data  $Y2$  and  $X2$ .

**NOTE:**

\*1: To determine pen or finger touch, the pressure of the touch needs to be determined. Generally, it is not necessary to have very high performance for this test; therefore, the 8-bit resolution mode is recommended (however, calculations will be shown here are in 12bit resolution mode).

$R_{X\text{-plate}}$ : Total X-axis resistor value (about 200Ω~ 600Ω)

$R_{Y\text{-plate}}$ : Total Y-axis resistor value (about 200Ω~ 600Ω)

X-Position: X-axis voltage sample value

Y-Position: Y-axis voltage sample value

Z1, Z2: Z1, Z2 voltage sample value

Z3, Z4: Z3, Z4 voltage sample value

X2, Y2: X2, Y2 voltage sample value

### 24.2.8 ADC VBAT Data Register (ADVDAT)

The read-only ADVDAT is a 16-bit register, it keep the sample data of VBAT. 0~11 bits are data.

| ADBDAT |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x1007001C |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | VDATA |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0     | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                       | RW |
|-------|----------|-----------------------------------------------------------------------------------|----|
| 15:12 | Reserved | Writing has no effect, read as zero.                                              | R  |
| 11:0  | VDATA    | Data of VBAT A/D convert.<br>When write to the register, DATA will be clear to 0. | RW |

#### Notes:

1 The measured voltage VBAT can directly calculate as follows:

$$V_{\text{BAT}} = \frac{\text{VDATA}}{4096} \cdot 1.2\text{V}$$

2 The better accurate voltage can be calculate as follow formula:

$$V_{\text{bat}} = (\text{slope1}/5000 + 0.2882) \cdot \text{VDATA} + \text{intercept1}/10 + 9.62$$

$$V_{\text{bat}} = (\text{slope2}/5000 + 0.2885) \cdot \text{VDATA} + \text{intercept2}/10 + 7.31$$

slope1 stored in efuse offset address 0x213, and this is a signed value

intercept1 stored in efuse offset address 0x212, this also a signed value

slope2 and intercept2 like slope1/intercept1, there are also signed value, but stored address is efuse offset 0x21b and 0x21a.

### 24.2.9 ADC AUX Data Register (ADADAT)

The read-only ADADAT is a 16-bit register, it keep the sample data. 0~11 bits are data.



| ADSDAT |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10070020 |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | ADATA |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0     | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                          | RW |
|-------|----------|----------------------------------------------------------------------|----|
| 15:12 | Reserved | Writing has no effect, read as zero.                                 | R  |
| 11:0  | ADATA    | Data of AUX.<br>When write to the register, DATA will be clear to 0. | RW |

The measured voltage  $V_{AUX}$  ( $V_{AUX1}$  and  $V_{AUX2}$ ) is as follows:

$$V_{SAD} = \frac{ADATA}{4096} \cdot AVDD33$$

#### 24.2.10 ADC Clock Divide Register (ADCLK)

The register ADCLK is used to set the A/D's clock dividing number.

| ADCLK |           |    |    |    |    |    |    |    |    |    |    |    |           |    |    | 0x10070028 |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|-----------|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31        | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19        | 18 | 17 | 16         | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | CLKDIV_MS |    |    |    |    |    |    |    |    |    |    |    | CLKDIV_US |    |    |            | CLKDIV |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST   | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0         | 1  | 0  | 0          | 0      | 0  | 0  | 1  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |

| Bits  | Name      | Description                                                                                                                                                                                                    | RW |
|-------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:16 | CLKDIV_MS | Dividing number to get ms clock from ADC clock.<br>$ms\_clk = us\_clk / (CLK\_MS + 1)$                                                                                                                         | RW |
| 15:8  | CLKDIV_US | Dividing number to get us clock from ADC clock.<br>$us\_clk = adc\_clk / (CLKDIV\_US + 1)$<br>$0 \leq CLKDIV\_10 \leq 127$                                                                                     | RW |
| 7:0   | CLKDIV    | Dividing number to get ADC clock from device clock.<br>The A/D works at the frequency between 20KHz and 200KHz.<br>If $CLKDIV = N$ , Then the freq of $adc\_clk = dev\_clk / (N + 1)$ .<br>$0 \leq N \leq 255$ | RW |

### 24.2.11 ADC Command Register (ADCMD)

ADC Command Register ADCMD is used for write touch screen control command by software. Then, if the cmd\_sel\_r = 1, the controller will read ADCMD's command, then use command to control touch screen. The controller has 32x32 bit FIFO to store commands, the command format like this.

| ADCMD |     |    |     |    |    |    |       |       |       |       |       |       |          |         |         | 0x10070024 |            |          |         |         |         |       |       |       |       |         |         |         |       |       |       |      |
|-------|-----|----|-----|----|----|----|-------|-------|-------|-------|-------|-------|----------|---------|---------|------------|------------|----------|---------|---------|---------|-------|-------|-------|-------|---------|---------|---------|-------|-------|-------|------|
| Bit   | 31  | 30 | 29  | 28 | 27 | 26 | 25    | 24    | 23    | 22    | 21    | 20    | 19       | 18      | 17      | 16         | 15         | 14       | 13      | 12      | 11      | 10    | 9     | 8     | 7     | 6       | 5       | 4       | 3     | 2     | 1     | 0    |
|       | PIL |    | RPU |    |    |    | XPSUP | XNSUP | YPSUP | XPGRU | XNGRU | YNGRU | VREFNAUX | VREFNXN | VREFNXP | VREFNYN    | VREFPVDD33 | VREFPAUX | VREFPXN | VREFPXP | VREFPYP | XPADC | XNADC | YPADC | YNADC | WIPEADC | AUX2ADC | AUX1ADC | RPUWP | RPUPX | RPUYX | APIL |
| RST   | 0   | 0  | 0   | 0  | 0  | 0  | 0     | 0     | 0     | 0     | 0     | 0     | 0        | 0       | 0       | 0          | 0          | 0        | 0       | 0       | 0       | 0     | 0     | 0     | 0     | 0       | 0       | 0       | 0     | 0     | 0     | 0    |

| Bits  | Name       | Description                                                                       | RW |
|-------|------------|-----------------------------------------------------------------------------------|----|
| 31    | PIL        | Current used for pressure measurement.<br>0: Ip = 200μA(default)<br>1: Ip = 400μA | RW |
| 30:26 | RPU        | Internal Pull-up resistor for Pen Detection.                                      | RW |
| 25    | XPSUP      | XP to TPVDD control Switch. 0: open; 1:close.                                     | RW |
| 24    | XNSUP      | XN to TPGND control Switch. 0: open; 1:close.                                     | RW |
| 23    | YPSUP      | YP to TPVDD control Switch. 0: open; 1:close.                                     | RW |
| 22    | XPGRU      | XP to TPGND control Switch. 0: open; 1:close.                                     | RW |
| 21    | XNGRU      | XN to TPGND control Switch. 0: open; 1:close.                                     | RW |
| 20    | YNGRU      | YN to TPGND control Switch. 0: open; 1:close.                                     | RW |
| 19    | VREFNAUX   | ADC low voltage reference to AUX control switch. 0: open; 1: close.               | RW |
| 18    | VREFNXN    | ADC low voltage reference to XN control switch. 0: open; 1: close.                | RW |
| 17    | VREFNXP    | ADC low voltage reference to XP control switch. 0: open; 1: close.                | RW |
| 16    | VREFNYN    | ADC low voltage reference to YN control switch. 0: open; 1: close.                | RW |
| 15    | VREFPVDD33 | ADC high voltage reference to VDD control switch. 0: open; 1: close.              | RW |
| 14    | VREFPAUX   | ADC high voltage reference to AUX control switch. 0: open; 1: close.              | RW |
| 13    | VREFPXN    | ADC high voltage reference to XN control switch. 0: open; 1: close.               | RW |
| 12    | VREFPXP    | ADC high voltage reference to XP control switch. 0: open; 1: close.               | RW |
| 11    | VREFPYP    | ADC high voltage reference to YP control switch. 0: open; 1: close.               | RW |
| 10    | XPADC      | Use XP as ADC input channel control switch. 0: open; 1: close.                    | RW |
| 9     | XNADC      | Use XN as ADC input channel control switch. 0: open; 1: close.                    | RW |
| 8     | YPADC      | Use YP as ADC input channel control switch. 0: open; 1: close.                    | RW |
| 7     | YNADC      | Use YN as ADC input channel control switch. 0: open; 1: close.                    | RW |
| 6     | WIPEADC    | Use WIPE as ADC input channel control switch. 0: open; 1: close.                  | RW |
| 5     | AUX2ADC    | Use AUX2 as ADC input channel control switch. 0: open; 1: close.                  | RW |

|   |         |                                                                  |    |
|---|---------|------------------------------------------------------------------|----|
| 4 | AUX1ADC | Use AUX1 as ADC input channel control switch. 0: open; 1: close. | RW |
| 3 | RPUWP   | Connect WP to RPU control switch. 0: open; 1: close.             | RW |
| 2 | RPUXP   | Connect XP to RPU control switch. 0: open; 1: close.             | RW |
| 1 | RPUYP   | Connect YP to RPU control switch. 0: open; 1: close.             | RW |
| 0 | APIL    | Use inter current source control switch. 0: open; 1: close.      | RW |

## 24.3 SAR A/D Controller Guide

The following describes steps of using SAR-ADC.

### 24.3.1 Power Down Mode

- 1 Then initial value of ADENA.POWER is 1, and the state of SADC is in dower down state.
- 2 When you want to use SADC, you should first set ADENA.POWERON to 0 to power on SADC. And you should wait for at least 2ms, then you can enable Touch Screen, VBAT and AUX.
- 3 When you want to power down SADC to get lower power, you should disable Touch Screen, VBAT and AUX, and then set ADENA.POWER to 1.

### 24.3.2 A Sample Touch Screen Operation

(Pen Down → Sample some data of several points → Pen Up)

- 1 Set ADCTRL to 0x1f to mask all the interrupt of SADC.
- 2 Set DMA\_EN to choose whether to use DMA to read the sample data out or to use CPU to read the sample data out.
- 3 Set ADCFG.RPU to choose the Internal Pull-up resistor for Pen Detection.
- 4 Set ADCFG.WIRE\_SEL to choose 4-wire or 5-wire mode in pendown detect.
- 5 Set ADCFG.CMD\_SEL to choose use hardware inter command or software command control touch screen. If you want to use software command, you must write your command by ADCMD register. And if you want to use hardware inter command, please straight to set ADCFG.SPZZ.
- 6 Set ADCFG.SPZZ and ADCFG.XYZ to choose sample mode.
  - a  $X_s \rightarrow Y_s$  (Single-end X → Single-end Y).
  - b  $X_d \rightarrow Y_d$  (Differential X → Differential Y).
  - c  $X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d$  or  $X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d$  (Reference register ADCFG.SPZZ)  
(Differential X → Differential Y → Differential Z1 → Differential Z2 or  
Differential X → Differential Y → Differential Z3 → Differential Z4).
  - d  $X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d \rightarrow X2 \rightarrow Y2$  or  $X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d \rightarrow X2 \rightarrow Y2$  (Reference register bit SPZZ).
- 7 Set ADCFG.SNUM to choose one point sampling times.
- 8 Set ADCLK.CLKDIV, ADCLK.CLKDIV\_US and ADCLK.CLKDIV\_MS to set A/D clock frequency.

- 9 Set ADWAIT to decide the wait time of pen down and the interval time between sampling different points. This time delay is necessary because when pen is put down or pen position change, there should be some time to wait the pen down signal to become stable.
- 10 Set ADSAME to decide the interval time between repeated sampling the same point. User can repeat sampling one point to get the most accurate data.
- 11 Set ADCTRL.PENDM to 0 to enable the pen down interrupt of touch panel.
- 12 Set ADENA.TCHEN to 1 to start touch panel.
- 13 When pen down interrupt happened, you should set ADCTRL.PENDM to 1 and clear ADSTATE.PEND to close pen down interrupt. Then you should clear ADSTATE.PENDU and set ADCTRL.PENUM to 0 to enable pen up interrupt.
- 14 When pen down interrupt happened, the SARADC is sampling data. When ADSTATE.DTCH to 1, user must read the sample data from ADTCH. The SARADC will not sample the next point until the whole data of the one point are read (no matter by CPU or DMA). If ADCFG.XYZ is mode zero and mode one, user needs to read 1\*ADCFG.SNUM times to get the whole data. In mode two, user needs to read 2\*ADCFG.SNUM times to get the whole data. And in mode three, user needs to read 3\*ADCFG.SNUM times to get all data.
- 15 Repeat 14 till pen up interrupt happened.
- 16 When pen up interrupt happened, you should set ADCTRL.PENUM to 1 and clear ADSTATE.PENU. Then you should clear ADSTATE.PEND and set ADCTRL.PENDM to 0 to enable pen down interrupt.
- 17 Wait pen down interrupt and repeat from 13.
- 18 When you want to shut down the touch screen, user can set the ADENA.TCHEN to 0. If the last point is not sampled completely, user can abandon it.

### 24.3.3 SLEEP mode Sample Operation

- 1 If the register ADCLK have not been set before, you should set ADCLK.CLKDIV, ADCLK.CLKDIV\_US and ADCLK.CLKDIV\_MS to set A/D clock frequency.
- 2 Clear ADSTATE.SLP\_RDY, then you can set ADENA.SLP\_MD to 1. When ADSTATE.SLP\_RDY = 1, the Touch Screen is have entered the SLEEP mode.
- 3 After that you should clear ADSTATE.SLPEND and set ADCTRL.SLPENDM to 0 to enable “in SLEEP mode pen down interrupt” and mask all other interrupts. Then you can execute the SLEEP instruction to enter the SLEEP mode.
- 4 When “in SLEEP mode pen down interrupt” happened, it will switch from the SLEEP mode to NORMAL. Then you should set ADCTRL.SLPENDM to 1 and clear ADSTATE.SLPEND to close “in SLEEP mode pen down interrupt”. Clear ADSTATE.SLP\_RDY, and you should set ADENA.SLP\_MD to 0. When ADSTATE.SLP\_RDY = 1, the Touch Screen is have exited the SLEEP mode.
- 5 Then you can do any other operations.

#### 24.3.4 VBAT Sample Operation

- 1 Set ADCLK.CLKDIV, ADCLK.CLKDIV\_US and ADCLK.CLKDIV\_MS to set A/D clock frequency.
- 2 Set ADENA.VBATEN to 1 to enable the channel.
- 3 When ADSTATE.VRDY = 1, you can read the sample data from ADVDAT. And the VBATEN will be set to 0 auto.

#### 24.3.5 AUX Sample Operation

- 1 Set ADCFG.CMD to choose one CMD. ( AUX1 or AUX2)
- 2 Set ADCLK.CLKDIV, ADCLK.CLKDIV\_US and ADCLK.CLKDIV\_MS to set A/D clock frequency.
- 3 Set ADENA.AUXEN to 1 to enable the channel.
- 4 When ADSTATE.ARDY = 1, you can read the sample data from ADADAT. And the AUXEN will be set to 0 auto.

#### 24.3.6 Disable Touch Screen

- 1 When ADENA.TCHEN=1, ADENA.VBATEN=0, ADENA.AUXEN=0.
- 2 Set ADENA.TCHEN to 0.
- 3 Read ADENA.TCHEN till it is set to 0 by hardware, then Touch Screen is fully disabled.

#### 24.3.7 Multi-touch Operation

If you want to detect multi-touch, you should follow to steps as below.

- 1 Set ADENA.YYZ=11.
- 2 Set ADCTRL.PENDM to 0 to enable the pen down interrupt of touch panel.
- 3 Set ADCFG.TCHEN=1 to start touch panel.
- 4 When pen down interrupt happened, you should set ADCTRL.PENDM to 1 and clear ADSTATE.PEND to close pen down interrupt. Then you should clear ADSTATE.PENDU and set ADCTRL.PENUM to 0 to enable pen up interrupt.
- 5 When ADSTATE.DTCH to 1, you can read the sample data from ADTCH. The measured data recorded as  $X_{2_1}$ ,  $Y_{2_1}$ , you need to compare the measurement values of  $X_{2_1}$ ,  $Y_{2_1}$  and calibration values of  $X_2$ ,  $Y_2$ . If  $X_{2_1} < X_2$  and  $Y_{2_1} < Y_2$ , now is two points touch.
- 6 If the next measure is two points touch, the measured data recorded as  $X_{2_2}$ ,  $Y_{2_2}$ , you can compare the  $X_{2_1}$ ,  $y_{2_1}$  and  $X_{2_2}$ ,  $Y_{2_2}$ . If  $X_{2_2} > X_{2_1}$  and  $Y_{2_2} > y_{2_1}$ , the touch movement state is shrinkage; if  $X_{2_2} < X_{2_1}$  and  $Y_{2_2} < Y_{2_1}$ , the touch movement state is expand.

**NOTE:** Before in normally measurement ,you must calibration the values of  $X_2$  and  $Y_2$ . When you calibration the  $X_2$ ,  $Y_2$  value, you need a single point of touch the touch panel, record the measurements data. Then repeat these measurements at least three times, recording measure data is  $X_{2_1}$ ,  $X_{2_2}$ ,  $X_{2_3}$  and  $Y_{2_1}$ ,  $Y_{2_2}$ ,  $Y_{2_3}$ . You can use the formula (5) ,(6) to calculate the  $X_2$ ,  $Y_2$  value.

$$X_2 = \frac{X_{2_1} + X_{2_2} + X_{2_3}}{3} \quad (5)$$

$$Y_2 = \frac{Y_{2_1} + Y_{2_2} + Y_{2_3}}{3} \quad (6)$$

### 24.3.8 Use Software Command Operation

If you want to use software write command, you should follow to steps as below.

- 1 Set ADCFG.CMD\_SEL =1.
- 2 Read ADCMD register once, discard the read-in data. This reading purpose is to activate the command logic.
- 3 Write you command to ADCMD register.
- 4 Write 0x00000000 to ADCMD register, indicate the written command is end.
- 5 Set ADENA.TCHEN=1 to start touch panel.

**NOTE:** The RPU values are 6bits in SARADC, but in ADCMD register , it has 5 bits only. In fact, the lowest RPU bit circuit is given as a fixed value of 1.

### 24.3.9 Use 5-wire touch panel Operation

If you want to use 5-wrie touch panel, you should complete the following set.

- 1 Set ADCFG.WIRE\_SEL=1 to use 5-wire pendown detect.
- 2 Write all your commands to the command FIFO through ADCMD register.
- 3 Set ADENA.TCHEN=1 to start touch panel.

**NOTE:** In 5-wire mode, the ADCFG.SNUM will disable. The control logic will execution the control commands until the command value equals to 0x00000000.

### 24.3.10 Use External Touch Screen Controller Operation

If you want to use external touch screen controller, you should set ADCENA.TCHEN=0 and ADCENA.PEND=0, than you can use external touch screen controller freely.

**NOTE:** In this mode, all switches will open(default), but you can use VBAT or AUX sample operation by configure the appropriate register.

### 24.3.11 Use TSC to support keypad

SADC TSC function can apply to a keypad, if touch screen is not used. Suppose the keypad is a NxM matrix, where X direction has N key columns and Y direction has M key rows. Kij is used to indicate the key in ith column from left to right and jth row from bottom to top, where i=0~(N-1) and j=0~(M-1). Figure 24-1 is a 6x5 keypad circuit. The blue color is for X direction network and pink color is for Y. The networks are composed by resistors and metal line. These two networks should

be connected to SADC 4 pins: XP/XN/YP/YN as illustrated in the figure. The gray circle is the key. When no key pressing, X network and Y network is open circuit. When a key is pressed, the X network and Y network is shorted under the key position.

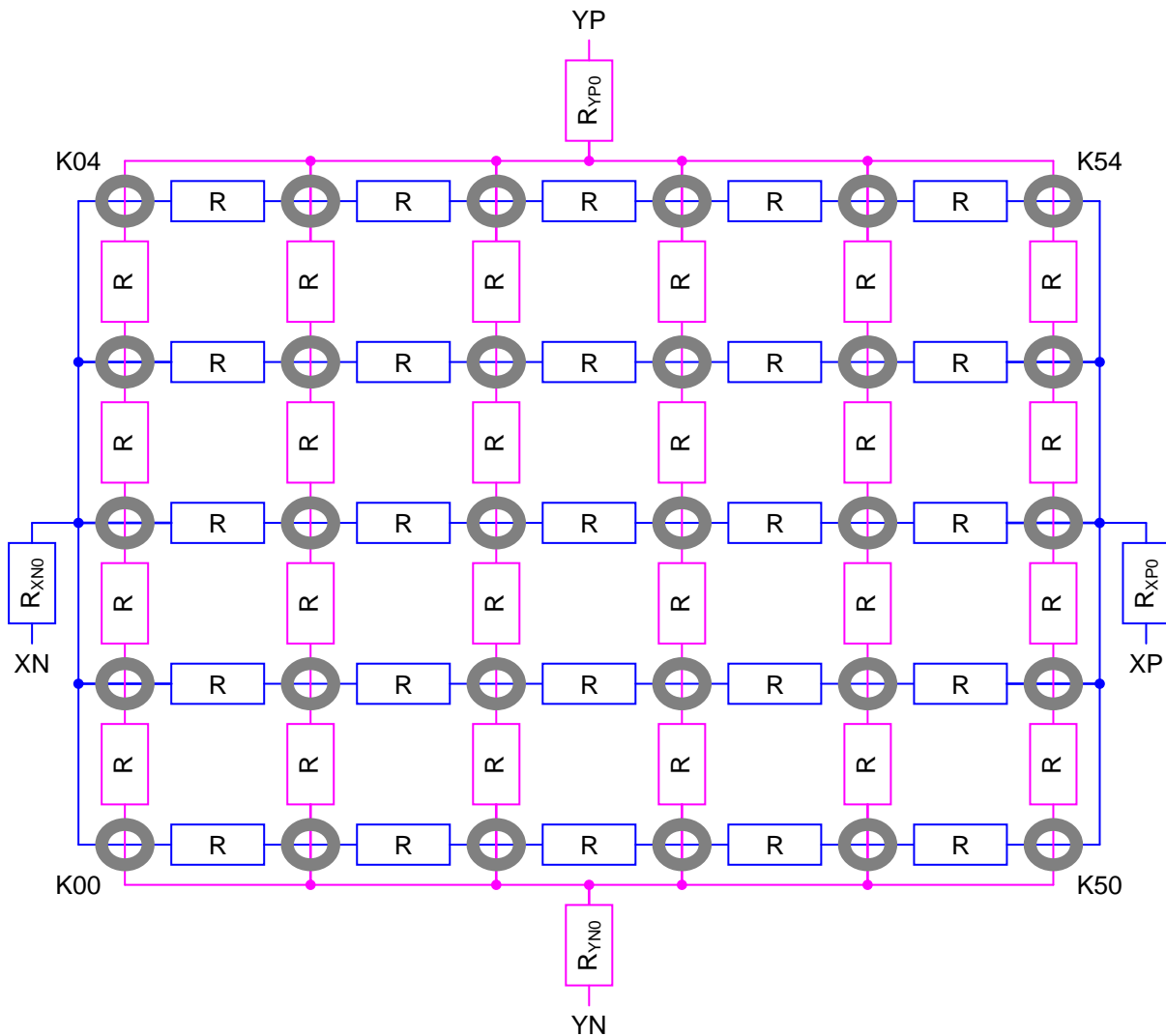


Figure 24-1 6x5 keypad circuit

When SADC is in waiting for pen-down status ( $C=1100$ ), the equivalent circuit is show in Figure 24-2. When the key is not pressed, XP is open and the PEN is pulled to VDDADC, which is logic 1. When the key Kij is pressed, the circuit is:  $VDDADC \rightarrow (10k\Omega \text{ resistor}) \rightarrow R_{XP} \rightarrow R_{YN} \rightarrow VSSADC$ .

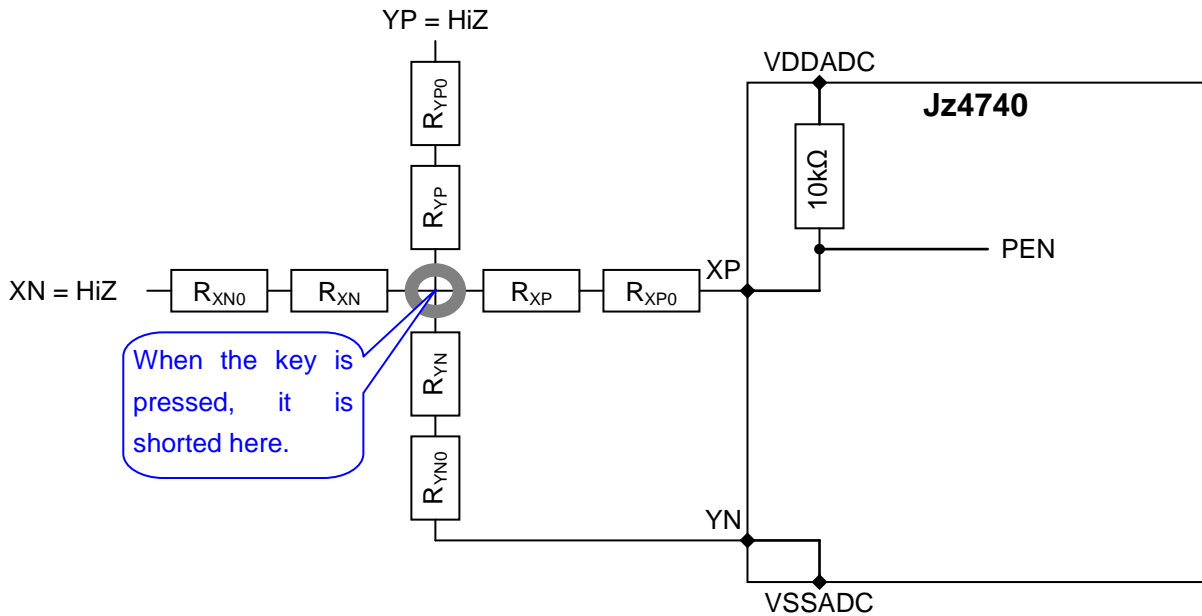


Figure 24-2 Wait for pen-down (C=1100) circuit

Where

$$R_{XP} = \frac{(N-1)^2 - i^2}{M \times (N-1-i) + 2i} \times R$$

$$R_{YN} = \frac{j \times (2M - 2 - j)}{N \times j + 2M - 2 - 2j} \times R$$

To ensure logic 0 at PEN in this case, following formula should be obeyed.

$$R_{XP} + R_{YN} + R_{XP0} + R_{YN0} \leq 3k\Omega \quad (7)$$

It is suggested the value of N and M is as close to each other as possible. For N=2~20, M=2~20 and M=(N-1, N or N+1), we found

$$R_{XP} + R_{YN} < 2.7 \times R \quad (8)$$

After key pressing is found, the key Kij location, columns and row, should be measured by using C=0010 and C=0011 respectively. The equivalent circuits are show in Figure 24-3 and Figure 24-4, where

$$R_{X0} = \frac{N-1}{M-1} \times R$$

$$R_{Y0} = \frac{M-1}{N-1} \times R$$

$$R_{XNi} = i \times R$$

$$R_{XPi} = (N-1-i) \times R$$

$$R_{YNj} = j \times R$$

$$R_{YPj} = (M-1-j) \times R$$



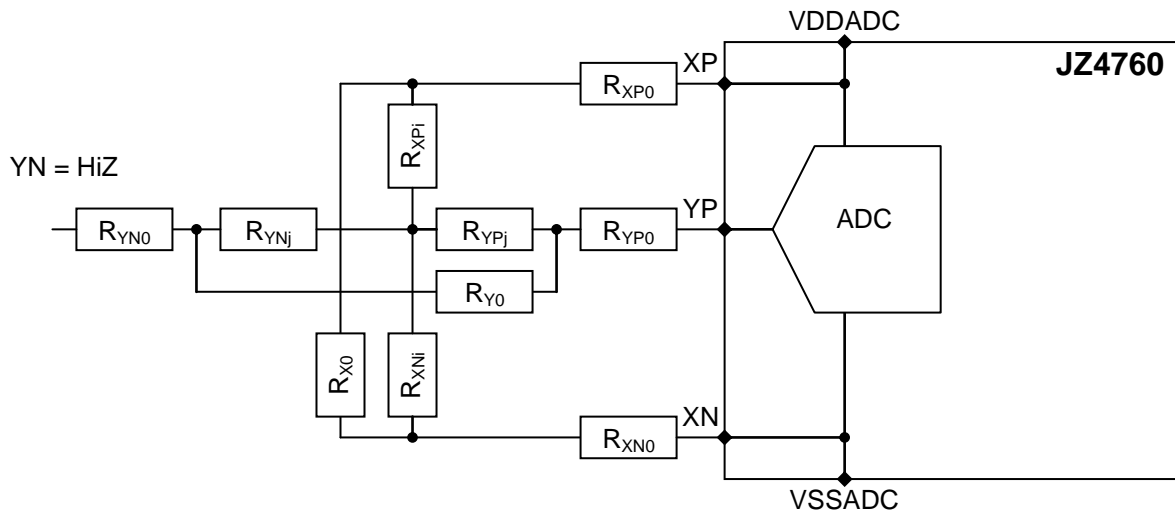


Figure 24-3 Measure X-position (C=0010) circuit

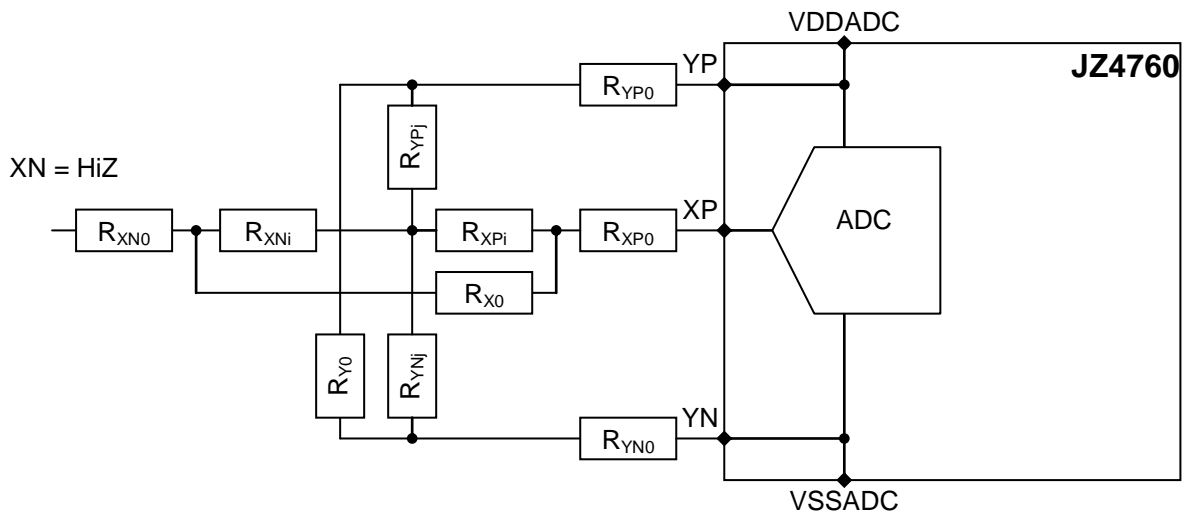


Figure 24-4 Measure Y-position (C=0011) circuit

So for Kij pressing, we should get ADC converted number Ni and Nj for i and j respectively.

$$N_i = \frac{R_{XN0} + \frac{i}{M} R}{R_{XN0} + \frac{N-1}{M} R + R_{XP0}} \times 4096$$

$$N_j = \frac{R_{YN0} + \frac{j}{N} R}{R_{YN0} + \frac{M-1}{N} R + R_{YP0}} \times 4096$$

It is required the resistor between XP and XN in case of C=0010, between YP and YN in case of C=0011, must be  $\geq 200\Omega$  and it better be  $\geq 500\Omega$ . Also consider the requirement in formula (7) and (8) above, we suggest to put  $R_{XP0} = R_{XN0} = R_{YP0} = R_{YN0} = 50\Omega$  or  $100\Omega$ , put  $R = 500\Omega \sim 1k\Omega$ .

To use the keypad, the software should set:

ADENA.TCHEN = 1

ADCFG.XYZ = 10

The operation is similar to touch screen.

## 25 Real Time Clock

### 25.1 Overview

The Real-Time Clock (RTC) unit can be operated in either chip main power is on or the main power is down but the RTC power is still on. In this case, the RTC power domain consumes only a few micro watts power.

The RTC contains a 32768Hz oscillator, the real time and alarm logic, and the power down and wakeup control logic.

#### 25.1.1 Features

RTC module has following features:

- Need external 32768Hz oscillator for 32k clock generation
- RTCLK selectable from the oscillator or from the divided clock of EXCLK, so that 32k crystal can be absent if the hibernating mode is not needed
- 32-bits second counter
- Programmable and adjustable counter to generate accurate 1 Hz clock
- Alarm interrupt, 1Hz interrupt
- Stand alone power supply, work in hibernating mode
- Power down controller
- Alarm wakeup
- External pin wakeup with up to 2s glitch filter

#### 25.1.2 Signal Descriptions

RTC has 5 signal IO pins and 2 power pin. They are listed and described in.

| Pin Names | Pin Loc | IO | IO Cell Char. | Pin Description                                                                   | Power  |
|-----------|---------|----|---------------|-----------------------------------------------------------------------------------|--------|
| RTCLK     |         | AI | 32768Hz       | RTCLK: 32768 clock input or OSC input                                             | VDDRTC |
| NC        |         | AO |               | NC: Not Connect                                                                   | -      |
| PWRON     |         | O  | 8mA           | PWRON: Power on/off control of main power                                         | VDDRTC |
| WKUP_     |         | I  | Schmitt       | WKUP_: Wake signal after main power down                                          | VDDRTC |
| PPRST_    |         | I  | Schmitt       | PPRST_: RTC power on reset and RESET-KEY reset input                              | VDDRTC |
| VDDRTC    |         | P  |               | VDDRTC: 1.8V power for RTC and hibernating mode controlling that never power down | -      |
| VDDRTC11  |         | P  |               | VDDRTC11:1.1V power for RTC core that never power down                            |        |

| Pin Names      | Pin Loc | IO      | IO Cell Char.         | Pin Description                                                                                                | Power      |
|----------------|---------|---------|-----------------------|----------------------------------------------------------------------------------------------------------------|------------|
| CLK32K<br>PD14 |         | O<br>IO | 8mA<br>pullup-pe      | 32768Hz clock output<br>PD14: GPIO group D bit14. When main power down, this pin is controlled by RTC register | VDDRT<br>C |
| OWI<br>PF30    |         | I<br>IO | 8mA,<br>Open-Dra<br>w | OWI: One wire interface<br>PF: GPIO group F bit 30                                                             | VDDRT<br>C |

**RTCLK** pin. need input 32.768KHz rtc clock. If do not use any clock, hibernate mode will be NOT available any more, and the time will lose if power down.

**PWRON** pin: this pin is used to control the main power on/off. Output high voltage means on and low voltage means off.

**WKUP\_** pin: hibernating mode wakeup input. (Default low active)

**PPRST\_** pin: This pin should be set to low voltage only in two cases.

- When RTC power is turned on. (so that whole chip is power on)
- A RESET-KEY is pressed.

**CLK32PD14** pin: output 32.768KHz RTC clock and can used as GPIO.

**OWIPF30** pin: output to reset PMU when hibernate wakeup and can used as GPIO.

## 25.2 Register Description

Table 25-1 Registers for real time clock

| Name   | Description               | RW | Reset Value                | Address    | Access Size |
|--------|---------------------------|----|----------------------------|------------|-------------|
| RTCCR  | RTC Control Register      | RW | 0x00000081 <sup>*1*2</sup> | 0x10003000 | 32          |
| RTCSR  | RTC Second Register       | RW | 0x????????                 | 0x10003004 | 32          |
| RTCSAR | RTC Second Alarm Register | RW | 0x????????                 | 0x10003008 | 32          |
| RTCGR  | RTC Regulator Register    | RW | 0x0????????                | 0x1000300C | 32          |

### NOTES:

- 1 <sup>\*1</sup>: Unless otherwise stated, the reset value is for PPRST\_ and Hibernating wakeup reset. WDT reset doesn't change the value.
- 2 <sup>\*2</sup>: The reset value can be either of 0x00000081, 0x00000091, 0x00000089, 0x00000099.

Table 25-2 Registers for hibernating mode

| Name  | Description                                        | RW | Reset Value              | Address    | Access Size |
|-------|----------------------------------------------------|----|--------------------------|------------|-------------|
| HCR   | Hibernate Control Register                         | RW | 0x00000000 <sup>*1</sup> | 0x10003020 | 32          |
| HWFCR | Wakeup filter counter Register in Hibernate mode   | RW | 0x0000????               | 0x10003024 | 32          |
| HRCR  | Hibernate reset counter Register in Hibernate mode | RW | 0x00000800               | 0x10003028 | 32          |

|         |                                           |    |                          |            |    |
|---------|-------------------------------------------|----|--------------------------|------------|----|
| HWCR    | Wakeup control Register in Hibernate mode | RW | 0xD2AD2D28* <sup>1</sup> | 0x1000302C | 32 |
| HWRSR   | Wakeup Status Register in Hibernate mode  | RW | 0x00000000* <sup>1</sup> | 0x10003030 | 32 |
| HSPR    | Scratch pattern register                  | RW | 0x????????               | 0x10003034 | 32 |
| WENR    | Write enable pattern register             | RW | 0x00000000               | 0x1000303C | 32 |
| CKPCR   | Configure the CLK32K pin value            | RW | 0x00000010               | 0x10003040 | 32 |
| OWIPCR  | Configure the OWI pin value               | RW | 0x00000010               | 0x10003044 | 32 |
| PWRONCR | Configure the power on mode               | RW | 0x????????               | 0x10003048 | 32 |

**NOTE:**

\*<sup>1</sup>: Unless otherwise stated, the reset value is for PPRST\_ and Hibernating wakeup reset. WDT reset doesn't change the value.

All these registers, include those for real time clock and for hibernating mode control, except otherwise stated, are implemented in RTCLK clock domain. When write to these registers, it needs about 1 ~ 2 RTCLK cycles to actually change the register's value and needs another RTCLK cycle to allow the next write access. A bit RTCCR.WRDY is used to indicate it. When RCR.WRDY is 1, it means the previous write is finished, a right value can be read from the target register, and a new write access can be issued. So before any write access, please make sure RCR.WRDY = 1.

### 25.2.1 RTC Control Register (RTCCR)

RTCCR contains bits to configure the real time clock features. Unless otherwise stated, the reset value is for PPRST\_ and Hibernating wakeup reset. WDT reset doesn't change the value.

| RTCCR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10003000 |      |     |       |    |     |    |        |      |   |   |   |   |   |   |   |   |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|------|-----|-------|----|-----|----|--------|------|---|---|---|---|---|---|---|---|
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15   | 14  | 13    | 12 | 11  | 10 | 9      | 8    | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | WRDY | 1HZ | 1HZIE | AF | AIE | AE | SELEXC | RTCE |   |   |   |   |   |   |   |   |
| RST   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0    | 0   | 0     | 0  | 0   | 0  | 0      | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**NOTE:**

\*<sup>1</sup>: These bits are reset in all resets: PPRST\_ input pin reset, hibernating reset and WDT reset.

| Bits | Name     | Description                                                                                                                                                                                                                                                                                       | RW |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:7 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                              | R  |
| 7    | WRDY     | Write ready flag. It is 0 when a write is currently processing and the value has not been written to the writing target register. No write to any RTC registers can be issued in this case, or the result is undefined. The read value from the target register is also undefined. The reading is | R  |

|        |                                                                       | meaningful and another write can be issued when it is 1. Please reference to descriptions in 25.2 for some more details. This bit is read only and write to it is ignored.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |             |   |                                                                       |   |                                                                     |    |
|--------|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------------|---|-----------------------------------------------------------------------|---|---------------------------------------------------------------------|----|
| 6      | 1HZ                                                                   | 1Hz flag. This bit is set by hardware once every 1 second through the 1Hz pulse if the real time clock is enabled (RTCCR.RTCE = 1). This bit can be cleared by software. Write 1 to this bit is ignored.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |             |   |                                                                       |   |                                                                     |    |
| 5      | 1HZIE                                                                 | 1Hz interrupt enable. Writing to this bit takes effect immediately without delay. <table border="1" data-bbox="485 566 1281 734"> <thead> <tr> <th>1HZIE</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1Hz interrupt is disabled.</td> </tr> <tr> <td>1</td> <td>1Hz interrupt is enabled. RTC issues interrupt when 1HZ bit is set.</td> </tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                | 1HZIE  | Description | 0 | 1Hz interrupt is disabled.                                            | 1 | 1Hz interrupt is enabled. RTC issues interrupt when 1HZ bit is set. | RW |
| 1HZIE  | Description                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 0      | 1Hz interrupt is disabled.                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 1      | 1Hz interrupt is enabled. RTC issues interrupt when 1HZ bit is set.   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 4      | AF                                                                    | Alarm flag. This bit is set by hardware when alarm match (RTCSR = RTCSAR) is found and alarm is enabled (RTCCR.AE = 1) and the real time clock is enabled (RTCCR.RTCE = 1). This bit can be cleared by software. Write 1 to this bit is ignored. Writing to this bit takes effect immediately.                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |             |   |                                                                       |   |                                                                     |    |
| 3      | AIE                                                                   | Alarm interrupt enable. <table border="1" data-bbox="485 987 1281 1155"> <thead> <tr> <th>AIE</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Alarm interrupt is disabled.</td> </tr> <tr> <td>1</td> <td>Alarm interrupt is enabled. RTC issues interrupt when AF is set.</td> </tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                            | AIE    | Description | 0 | Alarm interrupt is disabled.                                          | 1 | Alarm interrupt is enabled. RTC issues interrupt when AF is set.    | RW |
| AIE    | Description                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 0      | Alarm interrupt is disabled.                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 1      | Alarm interrupt is enabled. RTC issues interrupt when AF is set.      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 2      | AE                                                                    | Alarm enable. <table border="1" data-bbox="485 1205 1281 1330"> <thead> <tr> <th>AE</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Alarm function is disabled.</td> </tr> <tr> <td>1</td> <td>Alarm function is enabled.</td> </tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | AE     | Description | 0 | Alarm function is disabled.                                           | 1 | Alarm function is enabled.                                          | RW |
| AE     | Description                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 0      | Alarm function is disabled.                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 1      | Alarm function is enabled.                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 1      | SELEXC                                                                | The divided EXCLK is selected as RTCLK in rtc-hiber module. <table border="1" data-bbox="485 1375 1281 1585"> <thead> <tr> <th>SELEXC</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>OSC32K or RTCLK input clock is selected as RTCLK in rtc-hiber module.</td> </tr> <tr> <td>1</td> <td>The divided EXCLK is selected as RTCLK in rtc-hiber module.</td> </tr> </tbody> </table> <p>NOTE: If do not use any 32Khz clock (either input clock or using crystal), hibernate mode will be NOT available any more, and the time will lose if power down.<br/>CPM.OPCR.ERCS must be 0, when using SELEXC = 1.<br/>When the main chip power down, SELEXC will be 0 in internal circuit, in this time, RTCLK will use OSC32K clock.</p> | SELEXC | Description | 0 | OSC32K or RTCLK input clock is selected as RTCLK in rtc-hiber module. | 1 | The divided EXCLK is selected as RTCLK in rtc-hiber module.         | RW |
| SELEXC | Description                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 0      | OSC32K or RTCLK input clock is selected as RTCLK in rtc-hiber module. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 1      | The divided EXCLK is selected as RTCLK in rtc-hiber module.           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 0      | RTCE                                                                  | Real time clock enable. <table border="1" data-bbox="485 1879 1281 2004"> <thead> <tr> <th>RTCE</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Real time clock function is disabled.</td> </tr> <tr> <td>1</td> <td>Real time clock function is enabled.</td> </tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                             | RTCE   | Description | 0 | Real time clock function is disabled.                                 | 1 | Real time clock function is enabled.                                | RW |
| RTCE   | Description                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 0      | Real time clock function is disabled.                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |
| 1      | Real time clock function is enabled.                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |             |   |                                                                       |   |                                                                     |    |

### 25.2.2 RTC Second Register (RTCSR)

RTCSR is a 32-bit width second counter. It can be read and write by software. It is increased by 1 at every 1Hz pulse if the real time clock is enabled (RTCCR.RTCE = 1). When read, it should be read continued more than once and take the value if the adjacent results are the same. RTCSR is not initialized by any reset.

| RTCSR |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10003004 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31    | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | RTCSR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST   | ?     | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?          | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

### 25.2.3 RTC Second Alarm Register (RTCSAR)

RTCSAR serves as a second alarm register. Alarm flag (RTCCR.AF) is set to 1 when the RTCSR equals the RTCSAR in the condition of alarm is enabled (RTCCR.AE = 1) and the real time clock is enabled (RTCCR.RTCE = 1). RTCSAR can be read and write by software and is not initialized by any reset.

| RTCSAR |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10003008 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit    | 31     | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | RTCSAR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST    | ?      | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?          | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

### 25.2.4 RTC Regulator Register (RTCGR)

RTCGR is serves as the real time clock regulator, which is used to adjust the interval of the 1Hz pulse.

| RTCGR |      |                 |    |    |    |    |      |    |    |    |    |    |    |    |    | 0x1000300C |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|------|-----------------|----|----|----|----|------|----|----|----|----|----|----|----|----|------------|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31   | 30              | 29 | 28 | 27 | 26 | 25   | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | LOCK | Reserved        |    |    |    |    | ADJC |    |    |    |    |    |    |    |    |            | NC1HZ |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST   |      | 0 <sup>*1</sup> | 0  | 0  | 0  | 0  | 0    | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?          | ?     | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

**NOTE:**

\*1: This bit is reset in all resets: PPRST\_ input pin reset, hibernating reset and WDT reset.

| Bits  | Name                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW   |             |   |                            |   |                              |    |
|-------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------|---|----------------------------|---|------------------------------|----|
| 31    | LOCK                         | Lock bit. This bit is used to safeguard the validity of the data written into the RTCGR register. Once it is set, write to RTCGR is ignored. This bit can only be set by software and cleared by (any type of) resets. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>LOCK</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Write to RTCGR is allowed.</td> </tr> <tr> <td>1</td> <td>Write to RTCGR is forbidden.</td> </tr> </tbody> </table> | LOCK | Description | 0 | Write to RTCGR is allowed. | 1 | Write to RTCGR is forbidden. | RW |
| LOCK  | Description                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |             |   |                            |   |                              |    |
| 0     | Write to RTCGR is allowed.   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |             |   |                            |   |                              |    |
| 1     | Write to RTCGR is forbidden. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |             |   |                            |   |                              |    |
| 30:26 | Reserved                     | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                | R    |             |   |                            |   |                              |    |
| 25:16 | ADJC                         | This field specifies how many times it needs to add one 32kHz cycle for the 1Hz pulse interval in every 1024 1Hz pulses. In other word, among every 1024 1Hz pulses, ADJC number of them are triggered in every (NC1HZ + 2) 32kHz clock cycles, (1024 – ADJC) number of them are triggered in every (NC1HZ + 1) 32kHz clock cycles.                                                                                                                                                 | RW   |             |   |                            |   |                              |    |
| 15:0  | NC1HZ                        | This field specifies the number plus 1 of the working 32kHz clock cycles are contained in the 1Hz pulse interval. In other word, 1Hz pulse is triggered every (NC1HZ + 1) 32kHz clock cycles, if RTCGR.ADJC = 0.                                                                                                                                                                                                                                                                    | RW   |             |   |                            |   |                              |    |

### 25.2.5 Hibernate Control Register (HCR)

HCR contains the bit to control the main chip power on/off. Unless otherwise stated, the reset value is for PPRST\_ and Hibernating wakeup reset.

|            |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
|------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|---|
| <b>HCR</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x10003020</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |   |
| Bit        | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0  |   |
|            | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | PD |   |
| RST        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW |       |             |   |        |                               |   |     |                                    |    |
|------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-------|-------------|---|--------|-------------------------------|---|-----|------------------------------------|----|
| 31:1 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | R  |       |             |   |        |                               |   |     |                                    |    |
| 0    | PD       | Power down or power on bit. Besides writing by CPU, this bit will be set to 1 if an unknown reason main power supply off is detected. This bit controls the PWRON pin level. When co-working with some external components, this bit is used for power management of this chip. It is supposed when 1 is written to this bit, the main power supply of the chip, except RTC power, will be shut down immediately. After this bit is set to 1, all registers in RTC module, except RTCCR.1HZ and RTCCR.1HZIE, cannot be changed by write access. This bit is cleared by reset pin reset and hibernating reset. The later one is asserted by wakeup procedure. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>PD</th> <th>PWRON</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>VDDRTC</td> <td>No power down, keep power on.</td> </tr> <tr> <td>1</td> <td>0 V</td> <td>Power down enable, turn power off.</td> </tr> </tbody> </table> | PD | PWRON | Description | 0 | VDDRTC | No power down, keep power on. | 1 | 0 V | Power down enable, turn power off. | RW |
| PD   | PWRON    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |       |             |   |        |                               |   |     |                                    |    |
| 0    | VDDRTC   | No power down, keep power on.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |       |             |   |        |                               |   |     |                                    |    |
| 1    | 0 V      | Power down enable, turn power off.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |       |             |   |        |                               |   |     |                                    |    |



### 25.2.6 HIBERNATE mode Wakeup Filter Counter Register (HWFCR)

The HIBERNATE mode Wakeup Filter Counter Register (HWFCR) is a 32-bit read/write register .It filter the glitch generated by a dedicated wakeup pin. The HWFCR is not initialized by any reset.

| HWFCR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10003024 |       |    |    |    |    |    |          |   |   |   |   |   |   |   |   |   |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|-------|----|----|----|----|----|----------|---|---|---|---|---|---|---|---|---|
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15    | 14 | 13 | 12 | 11 | 10 | 9        | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | HWFCR |    |    |    |    |    | Reserved |   |   |   |   |   |   |   |   |   |
| RST   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | ?     | ?  | ?  | ?  | ?  | ?  | ?        | ? | ? | ? | ? | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                        | RW |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:16 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                               | R  |
| 15:5  | HWFCR    | Wakeup pin effective minimum time in number of 32 RTCLK cycles, used as glitch filter logic. Maximum of 2 seconds if the RTCLK is 32768Hz<br>If this value is configured to 0, and the pin keeps low longer than 15 RTCLK periods, it wakes up RTC from Hibernate. | RW |
| 4:0   | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                               | R  |

### 25.2.7 Hibernate Reset Counter Register (HRCR)

The Hibernate Reset Counter Register is a 32-bit read/write register that specifies hibernate reset assertion time. The HRCR is initialized by PPRST\_ .

| HRCR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10003028 |      |    |    |    |          |    |   |   |   |   |   |   |   |   |   |   |
|------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|------|----|----|----|----------|----|---|---|---|---|---|---|---|---|---|---|
| Bit  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15   | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|      | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | HRCR |    |    |    | Reserved |    |   |   |   |   |   |   |   |   |   |   |
| RST  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0    | 0  | 0  | 1  | 0        | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |

| Bits  | Name     | Description                                                                                                                                                                                                                      | RW |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:12 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                             | R  |
| 14:11 | HRCR     | HIBERNATE Reset waiting time. Number of 2048 RTCLK cycles. Default value 0x1, assert 125 ms. Maximum 1 second if the RTCLK is 32768Hz.<br>If this value is configured to 0, it will generate 2048 RTCLK HIBERNATE Reset, 62.5ms. | RW |
| 10:0  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                             | R  |

### 25.2.8 HIBERNATE Wakeup Control Register (HWCR)

The HIBERNATE Wakeup Control Register is a 32-bit read/write register that controls real time clock alarm wake up enable. The reset value only for PPRST\_.

| HWCR |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x1000302C |        |          |      |    |    |    |   |   |   |   |   |   |   |   |   |                 |
|------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|--------|----------|------|----|----|----|---|---|---|---|---|---|---|---|---|-----------------|
| Bit  | 31    | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15     | 14       | 13   | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0               |
|      | EPDET |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | WKUPVL | Reserved | EALM |    |    |    |   |   |   |   |   |   |   |   |   |                 |
| RST  | 1     | 1  | 0  | 1  | 0  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 1  | 1  | 0  | 1          | 0      | 0        | 1    | 0  | 1  | 1  | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 <sup>*1</sup> |

#### NOTE:

\*<sup>1</sup>: This bit is reset in PPRST\_ input pin reset and hibernating reset .

| Bits | Name     | Description                                                                          | RW |
|------|----------|--------------------------------------------------------------------------------------|----|
| 31:3 | EPDET    | Power detect enable.<br>0x1a55a5a5: disable(default)<br>Others: enable               | RW |
| 2    | WKUPVL   | RTC wakeup pin valid level.<br>0: Low level sensitive (default)<br>1: High sensitive | RW |
| 1    | Reserved | Writing has no effect, read as zero.                                                 | R  |
| 0    | EALM     | RTC Alarm wakeup enable.<br>0: disable<br>1: enable                                  | RW |

### 25.2.9 HIBERNATE Wakeup Status Register (HWRSR)

The HIBERNATE Wakeup Status Register is a 32-bit read/write register that reflects wakeup status bits.

| HWRSR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10003030 |     |          |    |     |          |     |     |   |   |   |                 |                 |   |   |   |   |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|-----|----------|----|-----|----------|-----|-----|---|---|---|-----------------|-----------------|---|---|---|---|
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15  | 14       | 13 | 12  | 11       | 10  | 9   | 8 | 7 | 6 | 5               | 4               | 3 | 2 | 1 | 0 |
|       | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | APD | Reserved | HR | PPR | Reserved | PIN | ALM |   |   |   |                 |                 |   |   |   |   |
| RST   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0   | 0        | 0  | 0   | 0        | 0   | 0   | ? | 0 | 0 | 1 <sup>*2</sup> | 1 <sup>*1</sup> | 0 | 0 | ? | ? |

#### NOTES:

\*<sup>1</sup>: This reset value only for PPRST\_. It is undefined in case of other resets.

\*<sup>2</sup>: This reset value only for HRST\_. It is undefined in case of other resets.

| Bits | Name                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW  |             |   |                                                                                        |   |                                                                                    |    |
|------|----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------------|---|----------------------------------------------------------------------------------------|---|------------------------------------------------------------------------------------|----|
| 31:9 | Reserved                                                                               | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | R   |             |   |                                                                                        |   |                                                                                    |    |
| 8    | APD                                                                                    | <p>Accident power down. When the software has not set to HIBERNATE state, the core power is down, then an accident power down is detected. APD is set and remains set until software clears it. This bit can only be written with 0. Write with 1 is ignored.</p> <table border="1"> <thead> <tr> <th>HR</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Accident power down has not occurred since the last time the software clears this bit.</td> </tr> <tr> <td>1</td> <td>Accident power down has occurred since the last time the software clears this bit.</td> </tr> </tbody> </table> | HR  | Description | 0 | Accident power down has not occurred since the last time the software clears this bit. | 1 | Accident power down has occurred since the last time the software clears this bit. | RW |
| HR   | Description                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |             |   |                                                                                        |   |                                                                                    |    |
| 0    | Accident power down has not occurred since the last time the software clears this bit. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |             |   |                                                                                        |   |                                                                                    |    |
| 1    | Accident power down has occurred since the last time the software clears this bit.     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |             |   |                                                                                        |   |                                                                                    |    |
| 7:6  | Reserved                                                                               | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | R   |             |   |                                                                                        |   |                                                                                    |    |
| 5    | HR                                                                                     | <p>Hibernate Reset. When a Hibernate reset detected, HR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 is ignored.</p> <table border="1"> <thead> <tr> <th>HR</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Hibernate reset has not occurred since the last time the software clears this bit.</td> </tr> <tr> <td>1</td> <td>Hibernate reset has occurred since the last time the software clears this bit.</td> </tr> </tbody> </table>                                                                        | HR  | Description | 0 | Hibernate reset has not occurred since the last time the software clears this bit.     | 1 | Hibernate reset has occurred since the last time the software clears this bit.     | RW |
| HR   | Description                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |             |   |                                                                                        |   |                                                                                    |    |
| 0    | Hibernate reset has not occurred since the last time the software clears this bit.     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |             |   |                                                                                        |   |                                                                                    |    |
| 1    | Hibernate reset has occurred since the last time the software clears this bit.         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |             |   |                                                                                        |   |                                                                                    |    |
| 4    | PPR                                                                                    | <p>PAD PIN Reset. When a PPRST_ is detected, PPR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 is ignored.</p> <table border="1"> <thead> <tr> <th>PPR</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>PPRST_ reset has not occurred since last time the software clears this bit.</td> </tr> <tr> <td>1</td> <td>PPRST_ reset has occurred since last time the software clears this bit.</td> </tr> </tbody> </table>                                                                                            | PPR | Description | 0 | PPRST_ reset has not occurred since last time the software clears this bit.            | 1 | PPRST_ reset has occurred since last time the software clears this bit.            | RW |
| PPR  | Description                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |             |   |                                                                                        |   |                                                                                    |    |
| 0    | PPRST_ reset has not occurred since last time the software clears this bit.            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |             |   |                                                                                        |   |                                                                                    |    |
| 1    | PPRST_ reset has occurred since last time the software clears this bit.                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |     |             |   |                                                                                        |   |                                                                                    |    |
| 3:2  | Reserved                                                                               | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | R   |             |   |                                                                                        |   |                                                                                    |    |
| 1    | PIN                                                                                    | Wakeup Pin Status bit. The bit is cleared when chip enters hibernating mode. It is set when exit the hibernating mode by wakeup pin. This bit can only be written with 0. Write with 1 is ignored.                                                                                                                                                                                                                                                                                                                                                                                                                  | RW  |             |   |                                                                                        |   |                                                                                    |    |
| 0    | ALM                                                                                    | RTC Alarm Status bit. The bit is cleared when chip enters hibernating mode. It is set when exit the hibernating mode by alarm. This bit can only be written with 0. Write with 1 is ignored.                                                                                                                                                                                                                                                                                                                                                                                                                        | RW  |             |   |                                                                                        |   |                                                                                    |    |

### 25.2.10 Hibernate Scratch Pattern Register (HSPR)

This is a scratch register used to hold a pattern. The software can check the pattern is kept to know whether RTC power has ever been down and whether it is needed to setup the real time clock.

|             |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>HSPR</b> |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x10003034</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>  | 31  | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             | PAT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>  | ?   | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                 | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description  | RW |
|------|------|--------------|----|
| 31:0 | PAT  | The pattern. | RW |

### 25.2.11 Write Enable Pattern Register (WENR)

This is a scratch register used to hold a pattern. The software can check the pattern is kept to know whether RTC power has ever been down and whether it is needed to setup the real time clock.

|             |     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------------|-----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>WENR</b> |     |          |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x1000303C</b> |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>  | 31  | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             | WEN | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |                   | WENPAT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>  |     | 0        | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                 | ?      | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits  | Name                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW  |             |   |                                                                       |   |                                     |   |
|-------|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------------|---|-----------------------------------------------------------------------|---|-------------------------------------|---|
| 31    | WEN                                                                   | <p>The write enable flag. If the WENPAT is 0xA55A then this bit will be 1.</p> <p>When the WEN changes to 1, the RTCCR, RTCSR, RTCSAR, RTCGR, HCR, HWFCR, HRCR, HWCR, HWRSR, HSPR registers could be changed.</p> <p>But RTCCR.SELEXC, RTCCR.HZIE, RTCCR.WRDY may change in any time.</p> <p>This bit is read only and write to it is ignored.</p> <p>This bit only reset by PPRST_ and HRST_.</p> <p>There is an exception, when system does NOT have RTC 32Khz crystal. MUST write 1 to RTCCR.SELEXC before write to any value to any other registers.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">WEN</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Other RTC registers is locked, write these registers will be ignored.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Other RTC registers can be changed.</td> </tr> </tbody> </table> | WEN | Description | 0 | Other RTC registers is locked, write these registers will be ignored. | 1 | Other RTC registers can be changed. | R |
| WEN   | Description                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |     |             |   |                                                                       |   |                                     |   |
| 0     | Other RTC registers is locked, write these registers will be ignored. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |     |             |   |                                                                       |   |                                     |   |
| 1     | Other RTC registers can be changed.                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |     |             |   |                                                                       |   |                                     |   |
| 30:16 | Reserved                                                              | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | R   |             |   |                                                                       |   |                                     |   |
| 15:0  | WENPAT                                                                | <p>The write enable pattern.</p> <p>Before writing any value to RTCCR, RTCSR, RTCSAR, RTCGR, HCR, HWFCR, HRCR, HWCR, HWRSR, HSPR registers, write 0xA55A to WENPAT to set these register writable. If this value is ok, WEN will</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | W   |             |   |                                                                       |   |                                     |   |

|  |  |                                                                                                                                          |  |
|--|--|------------------------------------------------------------------------------------------------------------------------------------------|--|
|  |  | <p>change to 1.</p> <p>But RTCCR.SELEXC and RTCCR.HZIE are writable in any time.</p> <p>These bits are write-only, always read as 0.</p> |  |
|--|--|------------------------------------------------------------------------------------------------------------------------------------------|--|

### 25.2.12 CLK32K Pin control register (CKPCR)

This is a CLK32K pin control register used to configure the CLK32K pin value. The CKPCR is initialized by PPRST\_.

| CKPCR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10003040 |    |    |    |    |    |    |   |   |   |   |   |        |          |          |         |       |   |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|---|---|---|---|---|--------|----------|----------|---------|-------|---|
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4      | 3        | 2        | 1       | 0     |   |
|       | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |   |   | CK32RD | CK32PULL | Reserved | CK32CTL | CK32D |   |
| RST   | ?        | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?          | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ?      | 1        | 0        | 0       | 0     | 0 |

| Bits | Name                                                                                                                               | Description                                                                                                                   | RW |                                                                                                                                                                                   |
|------|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:3 | Reserved                                                                                                                           | Writing has no effect, read as zero.                                                                                          | R  |                                                                                                                                                                                   |
| 5    | CK32RD                                                                                                                             | Read this bit will return CLK32K pin status.                                                                                  | R  |                                                                                                                                                                                   |
| 4    | CK32PULL                                                                                                                           | Pull up configures.                                                                                                           | RW |                                                                                                                                                                                   |
|      |                                                                                                                                    | <b>CK32PULL</b>                                                                                                               |    | <b>Description</b>                                                                                                                                                                |
|      |                                                                                                                                    | 0                                                                                                                             |    | Pull Up is enabled.                                                                                                                                                               |
| 1    | Pull Up is disabled.                                                                                                               |                                                                                                                               |    |                                                                                                                                                                                   |
| 3    | Reserved                                                                                                                           | Writing has no effect, read as zero.                                                                                          | R  |                                                                                                                                                                                   |
| 2:1  | CK32CTL                                                                                                                            | Output RTCLK to CLK32K pin.                                                                                                   | RW |                                                                                                                                                                                   |
|      |                                                                                                                                    | <b>CK32CTL</b>                                                                                                                |    | <b>Description</b>                                                                                                                                                                |
|      |                                                                                                                                    | 00                                                                                                                            |    | CLK32K pin is set to general input.<br>The pin value can be read by CK32RD bit or GPIO PD14 bit.<br>The input pin should not be left floating, if pull up (CK32PULL) is disabled. |
|      |                                                                                                                                    | 01                                                                                                                            |    | CLK32K pin is set to general output.<br>The pin output value is set by CK32D bit.                                                                                                 |
| 10   | GPIO PD14 controls CLK32K pin.<br>The pin output is set by GPIO PD14 bit.<br>The CLK32K pin is output CK32D bit in HIBERNATE mode. |                                                                                                                               |    |                                                                                                                                                                                   |
| 11   | Output RTCLK to CLK32K pin.<br>If this set, CLK32K pin will always output 32K clock, even in HIBERNATE mode.                       |                                                                                                                               |    |                                                                                                                                                                                   |
| 0    | CK32D                                                                                                                              | When CK32CTL is configured to general output or gpio (HIBERNATE), Write to this pin will output to CLK32K pin, if configured. | RW |                                                                                                                                                                                   |

### 25.2.13 OWI Pin control register (OWIPCR)

This is a OWI pin control register used to configure the OWI pin value. The OWIPCR is initialized by PPRST\_.

|               |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |       |         |          |        |      |    |   |   |   |   |   |   |   |   |   |   |
|---------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|-------|---------|----------|--------|------|----|---|---|---|---|---|---|---|---|---|---|
| <b>PWRPCR</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x10003044</b> |       |         |          |        |      |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                | 15    | 14      | 13       | 12     | 11   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|               | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   | OWIRD | OWIPULL | Reserved | OWICTL | OWID |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>    | ?        | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?                 | ?     | ?       | ?        | ?      | ?    | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits    | Name                                                                                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW      |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
|---------|------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------------|----|------------------------------------------------------------------------------------|----|-------------------------------------------------------------------------------|----|------------------------------------------------------------------------------------------------------------------------------------|----|---------------------------------------------------------------------------------------|----|
| 31:6    | Reserved                                                                                                                           | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | R       |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 5       | OWIRD                                                                                                                              | Read this bit will return OWI pin status.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | R       |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 4       | OWIPULL                                                                                                                            | Pull up configures.<br><table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">RETPULL</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Pull Up is enabled.</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Pull Up is disabled.</td> </tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RETPULL | Description | 0  | Pull Up is enabled.                                                                | 1  | Pull Up is disabled.                                                          | RW |                                                                                                                                    |    |                                                                                       |    |
| RETPULL | Description                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 0       | Pull Up is enabled.                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 1       | Pull Up is disabled.                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 3       | Reserved                                                                                                                           | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | R       |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 2:1     | OWICTL                                                                                                                             | Output one-wire control signal.<br><table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">OWICTL</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00</td> <td>OWI pin is set as input.<br/>The pin value can be read by OWI bit or GPIO PF30 bit.</td> </tr> <tr> <td style="text-align: center;">01</td> <td>OWI pin is set to general output.<br/>The pin output value is set by OWID bit.</td> </tr> <tr> <td style="text-align: center;">10</td> <td>GPIO PF30 controls OWI pin.<br/>The pin output is set by GPIO PF30 bit.<br/><b>The OWI pin is output OWID bit in HIBERNATE mode.</b></td> </tr> <tr> <td style="text-align: center;">11</td> <td>The OWI pin is set as input<br/>The pin value can be read by OWI bit or GPIO PF30 bit.</td> </tr> </tbody> </table> | OWICTL  | Description | 00 | OWI pin is set as input.<br>The pin value can be read by OWI bit or GPIO PF30 bit. | 01 | OWI pin is set to general output.<br>The pin output value is set by OWID bit. | 10 | GPIO PF30 controls OWI pin.<br>The pin output is set by GPIO PF30 bit.<br><b>The OWI pin is output OWID bit in HIBERNATE mode.</b> | 11 | The OWI pin is set as input<br>The pin value can be read by OWI bit or GPIO PF30 bit. | RW |
| OWICTL  | Description                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 00      | OWI pin is set as input.<br>The pin value can be read by OWI bit or GPIO PF30 bit.                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 01      | OWI pin is set to general output.<br>The pin output value is set by OWID bit.                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 10      | GPIO PF30 controls OWI pin.<br>The pin output is set by GPIO PF30 bit.<br><b>The OWI pin is output OWID bit in HIBERNATE mode.</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 11      | The OWI pin is set as input<br>The pin value can be read by OWI bit or GPIO PF30 bit.                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |
| 0       | OWID                                                                                                                               | When OWICTL is configured to general output or gpio (HIBERNATE),<br>Write to this pin will output to OWI pin, if configured.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW      |             |    |                                                                                    |    |                                                                               |    |                                                                                                                                    |    |                                                                                       |    |

### 25.2.14 Power on control register (PWRONCR)

This is a register used to control poweron signal. The PWRONCR is not initialized by any reset.

| PWRONCR |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x10003048 |                |    |    |    |          |    |   |       |   |   |   |   |   |   |   |   |
|---------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----------------|----|----|----|----------|----|---|-------|---|---|---|---|---|---|---|---|
| Bit     | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15             | 14 | 13 | 12 | 11       | 10 | 9 | 8     | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|         | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | PWROFF_L<br>EN |    |    |    | Reserved |    |   | PWRON |   |   |   |   |   |   |   |   |
| RST     | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0              | 0  | 0  | 0  | 0        | 0  | 0 | 0     | ? | ? | ? | ? | 0 | 0 | 0 | ? |

| Bits | Name       | Description                                                                                                                                           | RW |
|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:8 | Reserved   | Writing has no effect, read as zero.                                                                                                                  | R  |
| 7:4  | PWROFF_LEN | Power off assert length. Number of 4096 RTCLK cycles. Minimum 125ms and Maximum 2 second if the RTCLK is 32768Hz.                                     | RW |
| 3:1  | Reserved   | Writing has no effect, read as zero.                                                                                                                  | R  |
| 0    | PWRON_EN   | 0: disable<br>1: enable<br>If this bit set to 1, when PPRST_N assert, the PWRON will output low voltage, after PPRST_N. PWRON change to high voltage. | RW |

### 25.3 Time Regulation

Because of the inherent inaccuracy of crystal and other variables, the time counter may be inaccurate. This requires a slight adjustment. The application processor, through the RTCGR, lets you adjust the 1Hz time base to an error of less than 1ppm. Such that if the Hz clock were set to be 1Hz, there would be an error of less than 5 seconds per month.

To determine the value programmed into the RTCGR, you must first measure the output frequency at the oscillator multiplex (approximately 32 kHz) using an accurate time base, such as a frequency counter. This clock is externally visible by selecting the alternate function of GPIO.

To gain access to the clock, program this pin as an output and then switch to the alternate function. To trim the clock, divide the output of the oscillator by an integer value and fractional adjust it by periodically deleting clocks from the stream driving this integer divider.

After the true frequency of the oscillator is known, it must be split into integer and fractional portions. The integer portion of the value (minus one) is loaded into the NC1HZ field of the RTCGR.

The fractional part of the adjustment is done by periodically deleting clocks from the clock stream driving the Hz divider. The trim interval period is hardwired to be 1024 1Hz clock cycles (approximately 17 minutes). The number of clocks (represented by ADJC field of RTCGR) are deleted from the input clock stream per trim interval. If ADJC is programmed to be zero, then no trim operations occur and the RTC is clocked with the raw 32 kHz clock. The relationship between the Hz clock frequency and the nominal 32 kHz clock ( $f_1$  and  $f_{32K}$ , respectively) is shown in the following equation.

$$f1 = \frac{2^{10} \times (NC1HZ + 1)}{2^{10} \times (NC1HZ + 1) + ADJC} \times \frac{f32k}{NC1HZ + 1}$$

f1 = actual frequency of 1Hz clock

f32k = frequency of either 32.768KHz crystal output or 3.6864MHz crystal output further divided down to 32.914KHz

## 25.4 Operate Mode

### 25.4.1 Normal Mode

#### 25.4.1.1 Power Detect

When pprst reset, the HWCR.EPDET set to 0x1a55a5a5, and the power detect is disable. If you want To Enable the power detect, you should set HWCR.EPDET to other value, but not the 0x1a55a5a5. If power detect is enable, when CORE 1.2V lose, the RTC will into hibernate mode. And if power detect is not enable, the RTC can't enter hibernate mode when CORE 1.2V is lose.

#### 25.4.1.2 Power On Timing Diagram

The RTC support two power on procedure. If PWRONCR.PWRON\_EN is 0, the procedure like following figure, and when PWRONCR.PWRON\_EN value is 1, the power on procedure will like figure 25-2.

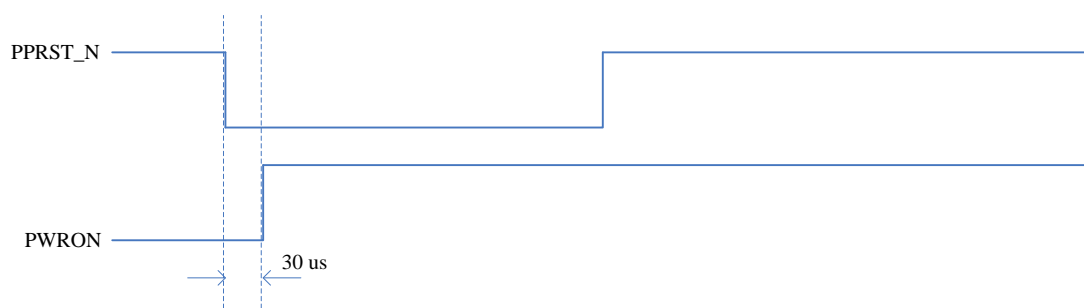


Figure 25-1 Core Power On directly

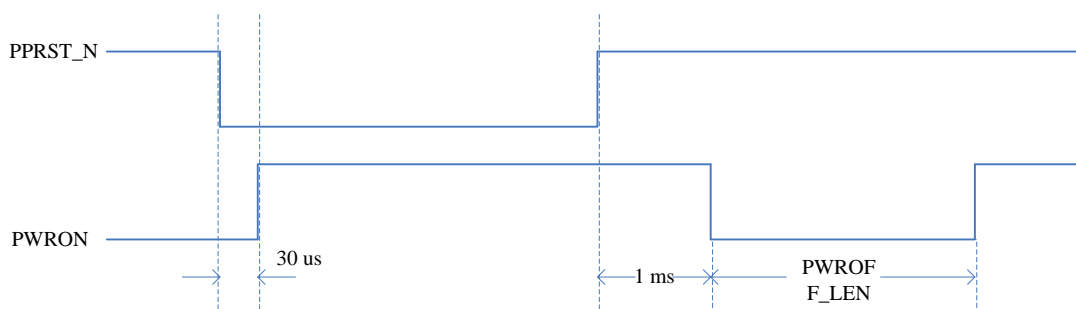


Figure 25-2 Core Power On-Off-On



**\*Note:** The register PWRONCR.PWROFF\_LEN indicate the power off length in above figure.

## 25.4.2 HIBERNATE Mode

The purpose of hibernate mode is to save the power. When in hibernate mode, CORE will no power consumption.

If you want to use hibernate mode, please first make sure RTCCR.SELEXC is 0.

When Software writes 1 to PD bit of HCR, the system at once enters HIBERNATE mode. The powers of CORE and IO are disconnected by PWRON pin, no power consumption to core and IO. When a wakeup event occurs, the core enters through a hibernate reset. Only CPM wake up logic and RTC is operating in HIBERNATE mode.

### 25.4.2.1 Procedure to Enter HIBERNATE mode

Before enter Normal HIBERNATE mode, software must complete following steps:

- 1 Finish the current operation and preserve all data to flash.
- 2 Configure the wake-up sources properly by configure HWCR.
- 3 Set HIBERNATE MODE. (Set PD bit in HCR to 1)

### 25.4.2.2 Procedure to Wake-up from HIBERNATE mode

- 1 The internal hibernate reset signal will be asserted if one of the wake-up sources is issued.
- 2 Check RSR to determine what caused the reset.
- 3 Check PIN/ALM bits of HWRSR in order to know whether or not the power-up is caused by which wake-up from HIBERNATE mode.
- 4 Configure the SDRAM memory controller.
- 5 Recover the data from flash.

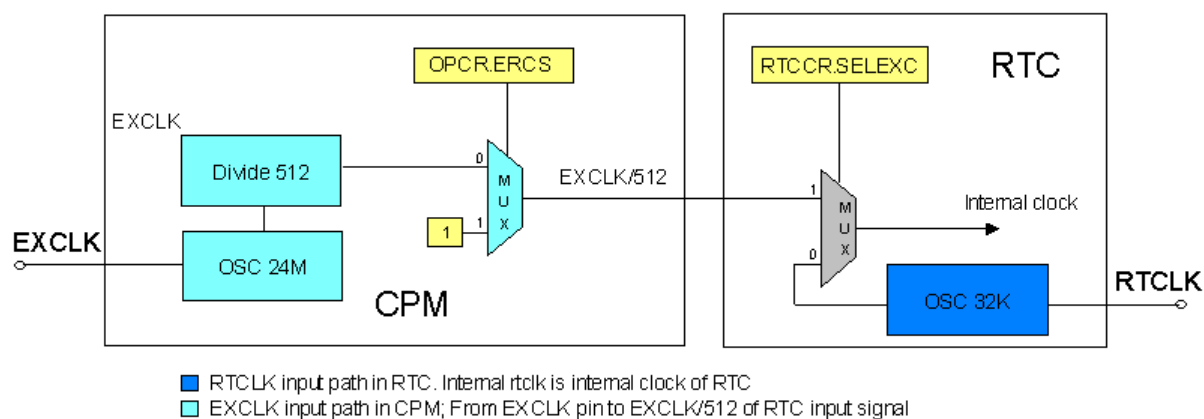
## 25.5 Clock select

There could be two clock input to RTC internal clock called rclk. One is OSC32k clock; the other is EXCLK/512.

The software MUST make sure the RTC run in valid clock configuration.

**Table 25-3 Clock select registers**

| RTCCR.SELEXC | CPM.ERCS | Description                      | Valid |
|--------------|----------|----------------------------------|-------|
| 0            | 0        | RTC use OSC32K clock.            | OK    |
| 0            | 1        |                                  | OK    |
| 1            | 0        | RTC use EXCLK/512 clock.         | OK    |
| 1            | 1        | RTC will lost clock. (Not Valid) | NO    |



**Figure 25-3 RTC clock selection path**

Changing RTCLK sequence:

- 1 There are both 32KHz crystal and 24Mhz EXCLK crystal connected, so RTCLK input path has 32Khz clock.

In this case, there is no need to change internal clock, so do NOT change SELEXC all the time.

- 2 There is no 32KHz crystal connected but only 24Mhz EXCLK crystal connected, so RTCLK input path has no clock.

In this case, should flow the sequence below to change internal clock:

- a Set OPCR.ERCS of CPM to 1; close EXCLK/512 to RTC.
- b Set CLKGR.RTC of CPM to 1; close PCLK to RTC.
- c Set RTCCR.SELEXC to 1; change internal clock to EXCLK/512.
- d Wait two clock period of clock.
- e Clear OPCR.ERCS of CPM to 0; open EXCLK/512 to RTC.
- f Clear CLKGR.RTC of CPM to 0; open PCLK to RTC.
- g Configure all RTC registers but RTCCR.SELEXC.
- h Check RTCCR.SELEXC == 1.
- i IF YES, finish this sequence; IF NO, do step (1) again.

**NOTE:** If using HIBERNATE mode, MUST have both 32KHz crystal (or input 32Khz clock) and 24Mhz EXCLK crystal connected, or RTC time will be insignificant.

## 26 EFUSE Slave Interface (EFUSE)

### 26.1 Overview

The EFUSE is provided 8K programmable bit, Total 8K bits are separated into seven segments, as below table show.

|       |        |        |         |      |         |         |
|-------|--------|--------|---------|------|---------|---------|
| 64bit | 128bit | 128bit | 3520bit | 8bit | 2296bit | 2048bit |
|-------|--------|--------|---------|------|---------|---------|

Each segment can be programmed separately or together.

Each segment has a protect bit, which has higher priority than program segment selection.

Programming frequency is wide, and programming time can adjust use the EFUSE register.

Initial value of EFUSE is 0, when programmed to 1, it doesn't able to program back to 0 anymore. Do not attempt to program any bit that already programmed to 1, such action will result unpredictable status to whole effuse block.

Programming voltage supply pin VDDQ:

- In program mode, supply VDDQ with 2.5V.  
Maximum accumulative time for VDDQ pin exposed under 2.5V+/-10% should be less than 1 sec.
- In read mode, leave VDDQ to 0V or floating.

### 26.2 Register Description

Following table lists all the register definitions. All registers' 32bit addresses are physical addresses.

**Table 26-1 EFUSE Register Description**

| Name     | Description              | RW | Reset Value | Address    | Access Size |
|----------|--------------------------|----|-------------|------------|-------------|
| EFUCTRL  | EFUSE Control Register   | RW | 0x001F0000  | 0x134100D0 | 32          |
| EFUCFG   | EFUSE Configure Register | RW | 0x00000000  | 0x134100D4 | 32          |
| EFUSTATE | EFUSE Status Register    | RW | 0x00000000  | 0x134100D8 | 32          |
| EFUDATA0 | EFUSE Data 0 Register    | RW | 0x000000??  | 0x134100DC | 32          |
| EFUDATA1 | EFUSE Data 1 Register    | RW | 0x00000000  | 0x134100E0 | 32          |
| EFUDATA2 | EFUSE Data 2 Register    | RW | 0x00000000  | 0x134100E4 | 32          |

|          |                       |    |            |            |    |
|----------|-----------------------|----|------------|------------|----|
| EFUDATA3 | EFUSE Data 3 Register | RW | 0x00000000 | 0x134100E8 | 32 |
| EFUDATA4 | EFUSE Data 4 Register | RW | 0x00000000 | 0x134100EC | 32 |
| EFUDATA5 | EFUSE Data 5 Register | RW | 0x00000000 | 0x134100F0 | 32 |
| EFUDATA6 | EFUSE Data 6 Register | RW | 0x00000000 | 0x134100F4 | 32 |
| EFUDATA7 | EFUSE Data 7 Register | RW | 0x00000000 | 0x134100F8 | 32 |

### 26.2.1 EFUSE Control Register (EFUCTRL)

|                |          |    |         |    |    |    |    |    |    |    |        |    |    |    |       |                   |    |    |    |    |    |    |   |   |   |   |       |       |   |   |   |   |
|----------------|----------|----|---------|----|----|----|----|----|----|----|--------|----|----|----|-------|-------------------|----|----|----|----|----|----|---|---|---|---|-------|-------|---|---|---|---|
| <b>EFUCTRL</b> |          |    |         |    |    |    |    |    |    |    |        |    |    |    |       | <b>0x134100D0</b> |    |    |    |    |    |    |   |   |   |   |       |       |   |   |   |   |
| Bit            | 31       | 30 | 29      | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21     | 20 | 19 | 18 | 17    | 16                | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5     | 4     | 3 | 2 | 1 | 0 |
|                | Reserved | CS | Address |    |    |    |    |    |    |    | Length |    |    |    | PG_EN | Reserved          |    |    |    |    |    |    |   |   |   |   | WR_EN | RD_EN |   |   |   |   |
| RST            | 0        | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 1  | 1  | 1  | 1     | 1                 | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0     | 0     | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                             | RW |
|-------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | Reserved | Writing has no effect, read as zero.                                                                                                                    | R  |
| 30    | CS       | EFUSE Bank select<br>0: select efuse low 4Kbit and operate it<br>1: select efuse high 4Kbit and operate it                                              | RW |
| 29:21 | Address  | Indicate the start address wants to program or read<br>Aligned to Byte                                                                                  | RW |
| 20:16 | Length   | Indicate the length wants to program or read, efuse operate length equals to this value add 1.<br>Byte number(min: 1byte, max: 32byte, default: 32byte) | RW |
| 15    | PG_EN    | Programming EFUSE enable<br>1: Enable 0: Disable                                                                                                        | RW |
| 14:2  | Reserved | Writing has no effect, read as zero.                                                                                                                    | RW |
| 1     | WR_EN    | Write data to EFUSE data buffer enable<br>1: Enable<br>0: Disable                                                                                       | RW |
| 0     | RD_EN    | Read data to EFUSE data buffer enable<br>1: Enable<br>0: Disable                                                                                        | RW |

### 26.2.2 EFUSE Configure Register (EFUCFG)

| EFUCFG |          |    |    |    |    |    |    |    |        |    |    |    |           |    |    | 0x134100D4 |        |    |    |    |           |    |   |   |   |   |   |   |   |   |   |   |
|--------|----------|----|----|----|----|----|----|----|--------|----|----|----|-----------|----|----|------------|--------|----|----|----|-----------|----|---|---|---|---|---|---|---|---|---|---|
| Bit    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22 | 21 | 20 | 19        | 18 | 17 | 16         | 15     | 14 | 13 | 12 | 11        | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|        | Reserved |    |    |    |    |    |    |    | RD_ADJ |    |    |    | RD_STROBE |    |    |            | WR_ADJ |    |    |    | WR_STROBE |    |   |   |   |   |   |   |   |   |   |   |
| RST    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0         | 1  | 0  | 0          | 0      | 0  | 0  | 0  | 0         | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name      | Description                                                                                                                                                          | RW |
|-------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | INT_EN    | Interrupt enable when read/write EFUSE finish.<br>1: Enable<br>0: Disable                                                                                            | RW |
| 30:25 | Reserved  | Writing has no effect, read as zero.                                                                                                                                 | R  |
| 23:20 | RD_ADJ    | Adjust number of h2clk cycles when EFUSE reading unsigned value, h2clk period multiply (RD_ADJ + 1) should greater than 6.5 ns.                                      | RW |
| 19:16 | RD_STROBE | Adjust number of h2clk cycles when EFUSE reading STROBE Signed value, h2clk period multiply(RD_ADJ + 5 + RD_STROBE) should greater than 35 ns, default 4.            | RW |
| 15:12 | WR_ADJ    | Adjust number of h2clk cycles when EFUSE programming unsigned value, h2clk period multiply (WR_ADJ + 1) should greater than 6.5 ns.                                  | RW |
| 11:0  | WR_STROBE | Adjust number of h2clk cycles when EFUSE programming STROBE Signed value, h2clk period multiply(WR_ADJ + 1666 + WR_STROBE) should between 9us and 11us, typical 10us | RW |

### 26.2.3 EFUSE Status Register (EFUSTATE)

| EFUSTATE |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x134100D8 |             |            |            |          |        |            |           |            |          |   |   |   |   |   |   |   |          |         |         |
|----------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|-------------|------------|------------|----------|--------|------------|-----------|------------|----------|---|---|---|---|---|---|---|----------|---------|---------|
| Bit      | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16         | 15          | 14         | 13         | 12       | 11     | 10         | 9         | 8          | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |          |         |         |
|          | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            | Gloabel_prt | CHIPID_prt | CustID_prt | Secwr_EN | PC_PRT | SecKey_prt | Secboot_e | Hdmike_prt | Reserved |   |   |   |   |   |   |   | HDMI_BUS | WR_DONE | RD_DONE |
| RST      | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0           | 0          | 0          | 0        | 0      | 0          | 0         | 0          | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 |          |         |         |

| Bits  | Name       | Description                                                                                                                                         | RW |
|-------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 30:16 | Reserved   | Writing has no effect, read as zero.                                                                                                                | R  |
| 15    | Global_PRT | Global protect bit, protect the protect bit segment.<br>1: protect, the protect segment can't write<br>0: no protect, the protect segment can write | R  |

|     |             |                                                                                                                                                                                               |    |
|-----|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 14  | CHIPID_PRT  | CHIP ID segment protect bit<br>1: protect, this segment can't write<br>0: no protect, this segment can write                                                                                  | R  |
| 13  | CustID_PRT  | Customer ID segment write protect bit<br>1: protect, this segment can't write<br>0: no protect, this segment can write                                                                        | R  |
| 12  | Secwr_EN    | Security boot write enable<br>1: enable, can write security boot enable bit<br>0: disable, can't write security boot enable bit                                                               | R  |
| 11  | PC_PRT      | CPU instructions segment write protect<br>1: protect, this segment can't write<br>0: no protect, this segment can write                                                                       | R  |
| 10  | HDMIKEY_PRT | HDMI KEY segment write protect bit<br>1: protect, this segment can't write<br>0: no protect, this segment can write                                                                           | R  |
| 9:  | Seckey_PRT  | Security Key segment write protect<br>1: protect, this segment can't write<br>0: no protect, this segment can write                                                                           | R  |
| 8   | Secboot_EN  | Security boot enable<br>1: enable Security boot<br>0: Normal boot                                                                                                                             | R  |
| 7:3 | Reserved    | Writing has no effect, read as zero.                                                                                                                                                          | RW |
| 2   | HDMI_BUSY   | HDMI controller read key process status<br>1: HDMI read key now, and the EFUSE controller is busy.<br>0: HDMI not read the key or the read process is finished, the EFUSE controller is idle. | R  |
| 1   | WR_DONE     | Write data to EFUSE operate done flag<br>1: done<br>0: not done<br>*This bit only can write to 0, and write to 0 will clear this flag.                                                        | RW |
| 0   | RD_DONE     | Read EFUSE data operate done flag<br>1: done<br>0: not done<br>*This bit only can write to 0, and write to 0 will clear this flag                                                             | RW |

### 26.2.4 EFUSE Data Register (EFUDATAn)

|     |                              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-----|------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|     | EFUDATAn (n=0,1,2,3,4,5,6,7) |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x134100DC~0x134100F8 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit | 31                           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|     |      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|     | DATA |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| RST | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name | Description                                                                                                | RW |
|------|------|------------------------------------------------------------------------------------------------------------|----|
| 31:0 | DATA | EFUSE DATA register.<br>*EFUSE DATA0 register mapped low operate address, others mapped increased address. | RW |

**NOTE:**

- The EFUSE 8K bits are mapped to chip io address, from 0x13410200 to 0x134105FF.
- The EFUSE 8K programmable bits are separated into seven segments as below table. The first Segment used to store 64bit random number, second segment store Ingenic chip ID, third segment used to store customer ID, forth Segment Reserved to Ingenic to program instructions, fifth segment store segment protect bit, sixth segment store HDMI key, and the last segment used to store Security boot key.

| Segment        | 1             | 2             | 3             | 4             | 5     | 6             | 7             |
|----------------|---------------|---------------|---------------|---------------|-------|---------------|---------------|
| Mapped address | 0x200 – 0x207 | 0x208 – 0x217 | 0x218 – 0x227 | 0x228 – 0x3DF | 0x3E0 | 0x3E1 – 0x4FF | 0x500 – 0x5FF |
| Bits           | 64bit         | 128bit        | 128bit        | 3520bit       | 8bit  | 2296bit       | 2048bit       |

- The protect bit segment store others segment write protect bit, if protect bit programmed to 1, corresponding segment can not be able to program. The protect bit map as below table

| The Protect bit Segment | Bit        | Mapped To Efuse Status Register |
|-------------------------|------------|---------------------------------|
|                         | 7          | Global_PRT                      |
|                         | 6          | CHIPID_PRT                      |
|                         | 5          | CustID_PRT                      |
|                         | 4          | Secwr_EN                        |
|                         | 3          | PC_PRT                          |
|                         | 2          | HDMIKEY_PRT                     |
|                         | 1          | SecKey_PRT                      |
| 0                       | Secboot_EN |                                 |

- The EFUSE can read or write maximum 256bit in each time.
- The first segment(store random number) and seventh segment(store security key) can read/write by SC-ROM only.
- We recommend read/write each segment separately.
- The address 0x218 to 0x21b are used to store SARADC calibration data, so please don't

program these two address.

## 26.3 Flow

### 26.3.1 Write EFUSE Flow

1. Set config register
2. Write want program data to EFUSE data buffer 0-7 registers
3. Set control register, indicate want to program address, data length.
4. Write control register PG\_EN bit to 1
5. Connect VDDQ pin to 2.5V
6. Write control register WR\_EN bit
7. Wait status register WR\_DONE set to 1.
8. Disconnect VDDQ pin from 2.5V.
9. Write control register PG\_EN bit to 0.

### 26.3.2 Write Security Key Flow

1. Write control register PG\_EN bit to 1
2. Connect VDDQ pin to 2.5V
3. Invoke SC-ROM controller to write data to Security key mapped address
4. Wait status register WR\_DONE set to 1
5. jump to step 3, program all data like this
6. Disconnect VDDQ pin from 2.5V.
7. Write control register PG\_EN bit to 0.

#### NOTE:

- \*<sup>1</sup>: Do NOT pre-charge VDDQ too early before fuse program started.  
The maximum 2.5V supply time to VDDQ must be strictly controlled less than 1sec.  
Use on-chip counter and GPIO to coordinate external supply source.
- \*<sup>2</sup>: The Security Key Segment can write use 1.3.2 flow only, The Random Number Segment will write in CP by Ingenic, others segment can write use 1.3.1 flow only.
- \*<sup>3</sup>: The SC-ROM Controller can write Security Key segment 32bits once time.
- \*<sup>4</sup>: Security Key Mapped address between 0x13410500 to 0x134105FF.
- \*<sup>5</sup>: Random Number Mapped address between 0x13410200 to 0x13410207.

### 26.3.3 Read EFUSE Flow

1. Set config register
2. Set control register to indicate what to read data address, read data numbers and read enable.
3. Wait status register RD\_DONE set to 1 or EFUSE interrupted, software can read EFUSE data buffer 0 – 8 registers.

### 26.3.4 Read Security Key/Random Number Flow

1. Invoke SC-ROM controller to read corresponding address.



**NOTE:**

\*<sup>1</sup>: EFSUE controller will read EFUSE protect bit segment automatically when chip reset, and update the efuse state register.

\*<sup>2</sup>: The Security Key and Random Number can read use 1.3.4 flow only, others segment can read use 1.3.3 flow only.

\*<sup>3</sup>: The SC-ROM Controller can read Security Key or Random Number 32bits once time.

## Section 6

# PERIPHERALS

## 27 TS Slave Interface (TSSI)

### 27.1 Overview

The TS Slave Interface (TSSI) is used to connect DTV Demodulator. It supports MPEG-2 Transport Stream (TS) as its input.

Features:

- Support both parallel mode and serial mode for TS data transfer
- TSDI0 or TSDI7 can be used to transfer data in serial mode
- The order of data in one byte supports LSB at first or MSB at first
- The order of data in one word supports LSB at first or MSB at first
- Input control signals and data can be either active high or active low
- Support using either positive or negative edge of TSCLK
- Support PID filtering function
- Up to 33 PID filters can be used when PID filtering function is enabled
- Support adding data 0 before or after transport stream
- Support master DMA transmission

### 27.2 Pin Description

**Table 27-1 TSSI Pin Description**

| Name    | I/O | Description             |
|---------|-----|-------------------------|
| TSDI0~7 | I   | TS data bus             |
| TSFAIL  | I   | TS packet uncorrectable |
| TSCLK   | I   | TS clock                |
| TSFRM   | I   | TS data valid           |
| TSSTR   | I   | TS packet start         |

### 27.3 Register Description

In this section, we will describe the registers in TSSI. Following table lists all the register definitions. All registers' 32bit addresses are physical addresses. And detailed function of each register will be described below.

The physical base address of TSSI controller is 0x134E0000.

Table 27-2 TSSI Register Description

| Name    | Description                    | RW | Reset Value | Offset Address | Access Size |
|---------|--------------------------------|----|-------------|----------------|-------------|
| TSENA   | TSSI Enable Register           | RW | 0x08        | 0x0000         | 8           |
| TSCFG   | TSSI Configure Register        | RW | 0x04FF      | 0x0004         | 16          |
| TSCTRL  | TSSI Control Register          | RW | 0x03        | 0x0008         | 8           |
| TSSTAT  | TSSI State Register            | RW | 0x00        | 0x000C         | 8           |
| TSFIFO  | TSSI FIFO Register             | R  | 0x????????  | 0x0010         | 32          |
| TSPEN   | TSSI PID Enable Register       | RW | 0x00000000  | 0x0014         | 32          |
| TSNUM   | TSSI Data Number Register      | RW | 0x00        | 0x0018         | 8           |
| TSDTR   | TSSI Data Trigger Register     | RW | 0x7F        | 0x001C         | 8           |
| TSPID0  | TSSI PID Filter Register 0     | RW | 0x00000000  | 0x0020         | 32          |
| TSPID1  | TSSI PID Filter Register 1     | RW | 0x00000000  | 0x0024         | 32          |
| TSPID2  | TSSI PID Filter Register 2     | RW | 0x00000000  | 0x0028         | 32          |
| TSPID3  | TSSI PID Filter Register 3     | RW | 0x00000000  | 0x002C         | 32          |
| TSPID4  | TSSI PID Filter Register 4     | RW | 0x00000000  | 0x0030         | 32          |
| TSPID5  | TSSI PID Filter Register 5     | RW | 0x00000000  | 0x0034         | 32          |
| TSPID6  | TSSI PID Filter Register 6     | RW | 0x00000000  | 0x0038         | 32          |
| TSPID7  | TSSI PID Filter Register 7     | RW | 0x00000000  | 0x003C         | 32          |
| TSPID8  | TSSI PID Filter Register 8     | RW | 0x00000000  | 0x0040         | 32          |
| TSPID9  | TSSI PID Filter Register 9     | RW | 0x00000000  | 0x0044         | 32          |
| TSPID10 | TSSI PID Filter Register 10    | RW | 0x00000000  | 0x0048         | 32          |
| TSPID11 | TSSI PID Filter Register 11    | RW | 0x00000000  | 0x004C         | 32          |
| TSPID12 | TSSI PID Filter Register 12    | RW | 0x00000000  | 0x0050         | 32          |
| TSPID13 | TSSI PID Filter Register 13    | RW | 0x00000000  | 0x0054         | 32          |
| TSPID14 | TSSI PID Filter Register 14    | RW | 0x00000000  | 0x0058         | 32          |
| TSPID15 | TSSI PID Filter Register 15    | RW | 0x00000000  | 0x005C         | 32          |
| TSDDA   | TSSI DMA Descriptor Address    | RW | 0x00000000  | 0x0060         | 32          |
| TSDTA   | TSSI DMA Target Address        | R  | 0x00000000  | 0x0064         | 32          |
| TSDID   | TSSI DMA Identifier            | R  | 0x00000000  | 0x0068         | 32          |
| TSDCMD  | TSSI DMA Command               | R  | 0x00000000  | 0x006C         | 32          |
| TSDST   | TSSI DMA Status                | RW | 0x00000000  | 0x0070         | 32          |
| TSTC    | TSSI Transfer Control Register | RW | 0x00000000  | 0x0074         | 32          |

### 27.3.1 TSSI Enable Register (TSENA)

The register TSENA is used to trigger TSSI to work.

| TSENA |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x0000  |          |   |      |       |        |        |     |
|-------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---------|----------|---|------|-------|--------|--------|-----|
| Bit   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7       | 6        | 5 | 4    | 3     | 2      | 1      | 0   |
|       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | SFT_RST | Reserved |   | FAIL | PEN_0 | PID_EN | DMA_EN | ENA |
| RST   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0       | 0        | 0 | 0    | 1     | 0      | 0      | 0   |

| Bits | Name     | Description                                                                                                                      | RW |
|------|----------|----------------------------------------------------------------------------------------------------------------------------------|----|
| 7    | SFT_RST  | TSSI FIFO software-reset. Set it to 1 and later it will be cleared by hardware auto.<br>0: Stop reset<br>1: Start reset          | RW |
| 6:5  | Reserved | Writing has no effect, read as zero.                                                                                             | R  |
| 4    | FAIL     | 0:FAIL signal is decided by TSSI Demodulator<br>1:FAIL signal is equal to 1                                                      |    |
| 3    | PEN_0    | Choose PID filter enable for PID=0.<br>0: not enable<br>1: enable                                                                | RW |
| 2    | PID_EN   | Enable / disable the PID filtering function.<br>0: PID filtering function is not enabled<br>1: PID filtering function is enabled | RW |
| 1    | DMA_EN   | Enable / disable the DMA mode.<br>0: DMA mode is not enabled (CPU only)<br>1: DMA mode is enabled                                | RW |
| 0    | ENA      | Enable / disable the TSSI module.<br>0: TSSI is not enabled<br>1: TSSI is enabled                                                | RW |

### 27.3.2 TSSI Configure Register (TSCFG)

The register TSCFG is used to configure the TSSI.

| TSCFG |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x0004  |          |          |        |        |        |       |          |       |         |         |         |          |   |   |   |   |   |
|-------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---------|----------|----------|--------|--------|--------|-------|----------|-------|---------|---------|---------|----------|---|---|---|---|---|
| Bit   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 16      | 15       | 14       | 13     | 12     | 11     | 10    | 9        | 8     | 7       | 6       | 5       | 4        | 3 | 2 | 1 | 0 |   |
|       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F_TRIGV | Reserved | TRANS_MD | EDN_WD | EDN_BT | TSDI_H | USE_0 | TSCLK_CH | PARAL | TSCLK_P | TSFRM_H | TSSTR_H | TSFAIL_H |   |   |   |   |   |
| RST   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0       | 0        | 0        | 0      | 0      | 1      | 0     | 0        | 1     | 1       | 1       | 1       | 1        | 1 | 1 | 1 | 1 | 0 |

| Bits  | Name    | Description                        | RW |
|-------|---------|------------------------------------|----|
| 16:14 | F_TRIGV | Specify the trigger value of FIFO. | RW |

|       |                      | <b>F_TRIGV</b>                                                                                                                                                  | <b>Description</b>                 |    |
|-------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|----|
|       |                      | 0                                                                                                                                                               | Trigger Value is 4                 |    |
|       |                      | 1                                                                                                                                                               | Trigger Value is 8                 |    |
|       |                      | 2                                                                                                                                                               | Trigger Value is 16                |    |
|       |                      | 3                                                                                                                                                               | Trigger Value is 32                |    |
|       |                      | 4                                                                                                                                                               | Trigger Value is 48                |    |
|       |                      | 5                                                                                                                                                               | Trigger Value is 64                |    |
|       |                      | 6                                                                                                                                                               | Trigger Value is 80                |    |
|       |                      | 7                                                                                                                                                               | Trigger Value is 96                |    |
|       |                      | Trigger value should be greater than Expected Transfer Size.                                                                                                    |                                    |    |
| 13:12 | Reserved             | Writing has no effect, read as zero.                                                                                                                            |                                    | R  |
| 11:10 | TRANS_MD             | Choose the mode of adding data 0.                                                                                                                               |                                    | RW |
|       |                      | <b>TRANS_MD</b>                                                                                                                                                 | <b>Description</b>                 |    |
|       |                      | 0                                                                                                                                                               | Add data 0 before transport stream |    |
|       |                      | 1                                                                                                                                                               | Add data 0 after transport stream  |    |
|       |                      | 2                                                                                                                                                               | Do not add data 0                  |    |
|       |                      | 3                                                                                                                                                               | Reserved                           |    |
| 9     | EDN_WD <sup>*1</sup> | The order of data in word.                                                                                                                                      |                                    | RW |
| 8     | EDN_BT <sup>*1</sup> | The order of data in byte.                                                                                                                                      |                                    | RW |
| 7     | TSDI_H               | Choose the polarity of TSDI0~7.<br>0: TSDI0~7 is active low<br>1: TSDI0~7 is active high                                                                        |                                    | RW |
| 6     | USE_0                | USE_0 is only used in SERIAL mode (TSCFG.PARAL=0).<br>0: Use TSDI7 to transfer data<br>1: Use TSDI0 to transfer data                                            |                                    | RW |
| 5     | TSCLK_CH             | Choose how to use TSCLK.<br>0: When $f_{pdk} > 3f_{TSCLK}$<br>1: When $f_{pdk} > 2f_{TSCLK}$                                                                    |                                    | RW |
| 4     | PARAL                | Choose the working mode of TSSI.<br>0: Serial Mode<br>1: Parallel Mode                                                                                          |                                    | RW |
| 3     | TSCLK_P              | This bit is used to determine which edge of TSCLK is used when TSSI is sampling data.<br>0: Use the negative edge of TSCLK<br>1: Use the positive edge of TSCLK |                                    | RW |
| 2     | TSFRM_H              | Choose the polarity of TSFRM.<br>0: TSFRM is active low<br>1: TSFRM is active high                                                                              |                                    | RW |
| 1     | TSSTR_H              | Choose the polarity of TSSTR.<br>0: TSSTR is active low<br>1: TSSTR is active high                                                                              |                                    | RW |
| 0     | TSFAIL_H             | Choose the polarity of TSFAIL.                                                                                                                                  |                                    | RW |

|  |                                                     |  |
|--|-----------------------------------------------------|--|
|  | 0: TSFAIL is active low<br>1: TSFAIL is active high |  |
|--|-----------------------------------------------------|--|

**NOTE:**

<sup>\*1</sup>: END\_BT and END\_WD in register TSCFG.

|                 |                 |                 |                 |                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Byte0<br>Bit0-7 | Byte1<br>Bit0-7 | Byte2<br>Bit0-7 | Byte3<br>Bit0-7 | Byte4<br>Bit0-7 | Byte5<br>Bit0-7 | Byte6<br>Bit0-7 | Byte7<br>Bit0-7 |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|

|          |       |       |       |       |
|----------|-------|-------|-------|-------|
| END_WD=0 | Byte0 | Byte1 | Byte2 | Byte3 |
|          | Byte4 | Byte5 | Byte6 | Byte7 |

|          |       |       |       |       |
|----------|-------|-------|-------|-------|
| END_WD=1 | Byte3 | Byte2 | Byte1 | Byte0 |
|          | Byte7 | Byte6 | Byte5 | Byte4 |

|          |      |      |      |      |      |      |      |      |
|----------|------|------|------|------|------|------|------|------|
| END_BT=0 | Bit0 | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 |
|----------|------|------|------|------|------|------|------|------|

|          |      |      |      |      |      |      |      |      |
|----------|------|------|------|------|------|------|------|------|
| END_BT=1 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|----------|------|------|------|------|------|------|------|------|

### 27.3.3 TSSI Control Register (TSCTRL)

The register TSCTRL is used to control TSSI to work.

|               |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  |               |         |         |   |   |   |   |   |   |
|---------------|--|--|--|--|--|--|--|--|--|--|--|--|----------|--|--|---------------|---------|---------|---|---|---|---|---|---|
| <b>TSCTRL</b> |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  | <b>0x0008</b> |         |         |   |   |   |   |   |   |
| <b>Bit</b>    |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  | 7             | 6       | 5       | 4 | 3 | 2 | 1 | 0 |   |
|               |  |  |  |  |  |  |  |  |  |  |  |  | Reserved |  |  | F_DTRM        | F_OVRNM | F_TRIGM |   |   |   |   |   |   |
| <b>RST</b>    |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  | 0             | 0       | 0       | 0 | 0 | 0 | 0 | 1 | 1 |

| Bits | Name     | Description                                                  | RW |
|------|----------|--------------------------------------------------------------|----|
| 7:3  | Reserved | Writing has no effect, read as zero.                         | R  |
| 2    | F_DTRM   | FIFO data trigger interrupt mask.<br>0: enabled<br>1: masked | RW |
| 1    | F_OVRNM  | FIFO overrun interrupt mask.<br>0: enabled<br>1: masked      | RW |

|   |         |                                                         |    |
|---|---------|---------------------------------------------------------|----|
| 0 | F_TRIGM | FIFO trigger interrupt mask.<br>0: enabled<br>1: masked | RW |
|---|---------|---------------------------------------------------------|----|

### 27.3.4 TSSI State Register (TSSTAT)

The register TSSTAT is used to keep the state of TSSI.

|               |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  |               |        |        |   |   |   |   |   |   |
|---------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----------|--|--|---------------|--------|--------|---|---|---|---|---|---|
| <b>TSSTAT</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  | <b>0x000C</b> |        |        |   |   |   |   |   |   |
| <b>Bit</b>    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  | 7             | 6      | 5      | 4 | 3 | 2 | 1 | 0 |   |
|               |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved |  |  | F_DTR         | F_OVRN | F_TRIG |   |   |   |   |   |   |
| <b>RST</b>    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  | 0             | 0      | 0      | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                     | RW |
|------|----------|-----------------------------------------------------------------|----|
| 7:3  | Reserved | Writing has no effect, read as zero.                            | R  |
| 2    | F_DTR    | FIFO data trigger interrupt flag.<br>1: active<br>0: not active | RW |
| 1    | F_OVRN   | FIFO overrun interrupt flag.<br>1: active<br>0: not active      | RW |
| 0    | F_TRIG   | FIFO trigger interrupt flag.<br>1: active<br>0: not active      | RW |

### 27.3.5 TSSI FIFO Register (TSFIFO)

The register TSFIFO is corresponded to TSSI FIFO.

|               |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|---------------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>TSFIFO</b> |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x0010</b> |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>    | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14            | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|               | DATA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>    | ?    | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?             | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

### 27.3.6 TSSI PID Enable Register (TSPEN)

The register TSPEN is used to control the PID filtering.



| TSPEN |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       | 0x0014 |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |
|-------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 31     | 30     | 29     | 28     | 27     | 26     | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16     | 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|       | EN_151 | EN_141 | EN_131 | EN_121 | EN_111 | EN_101 | EN_91 | EN_81 | EN_71 | EN_61 | EN_51 | EN_41 | EN_31 | EN_21 | EN_11 | EN_01  | EN_150 | EN_140 | EN_130 | EN_120 | EN_110 | EN_100 | EN_90 | EN_80 | EN_70 | EN_60 | EN_50 | EN_40 | EN_30 | EN_20 | EN_10 | EN_00 |
| RST   | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

| Bits  | Name              | Description                                                      | RW |
|-------|-------------------|------------------------------------------------------------------|----|
| 31:16 | EN_x1<br>(x=15~0) | PID filter enable for TSPIDx.PID1.<br>0: not enable<br>1: enable | RW |
| 15:0  | EN_x0<br>(x=15~0) | PID filter enable for TSPIDx.PID0.<br>0: not enable<br>1: enable | RW |

### 27.3.7 TSSI Data Number Register (TSNUM)

The register TSNUM is used to show the current number of the data in FIFO.

| TSNUM |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x0018   |      |   |   |   |   |   |   |  |  |  |
|-------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----------|------|---|---|---|---|---|---|--|--|--|
| Bit   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7        | 6    | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |
|       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved | DNUM |   |   |   |   |   |   |  |  |  |
| RST   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0        | 0    | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |

| Bits | Name     | Description                                                                           | RW |
|------|----------|---------------------------------------------------------------------------------------|----|
| 7    | Reserved | Writing has no effect, read as zero.                                                  | R  |
| 6:0  | DNUM     | The current number of data in FIFO.<br>The range of the data number is from 0 to 127. | RW |

### 27.3.8 TSSI Data Trigger Register (TSDTR)

The register TSDTR is used to trigger the FIFO level interrupt when the current data number in the FIFO is more than the value in TSDTR.

| TSDTR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x001C   |     |   |   |   |   |   |   |  |  |  |
|-------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----------|-----|---|---|---|---|---|---|--|--|--|
| Bit   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7        | 6   | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |
|       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved | DTR |   |   |   |   |   |   |  |  |  |

|     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |   |   |   |   |   |   |   |   |   |
|-----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|---|---|---|---|
| RST |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|-----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|---|---|---|---|

| Bits | Name     | Description                                                                   | RW |
|------|----------|-------------------------------------------------------------------------------|----|
| 7    | Reserved | Writing has no effect, read as zero.                                          | R  |
| 6:0  | DTRG     | The trigger number of FIFO.<br>The range of the data number is from 0 to 127. | RW |

### 27.3.9 TSSI PID Filter Registers (TSPID0~15)

The registers TSPID0~15 are used to store PID values that need to be filtered from MPEG-2 TS.

| TSPID0~15 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x0020~0x005C |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |  |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| Bit       | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16            | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |   |  |
|           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |  |
|           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |  |
| RST       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0             | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

| Bits  | Name     | Description                                  | RW |
|-------|----------|----------------------------------------------|----|
| 31:29 | Reserved | Writing has no effect, read as zero.         | R  |
| 28:16 | PID1     | Set the PID value that needs to be filtered. | RW |
| 15:13 | Reserved | Writing has no effect, read as zero.         | R  |
| 12:0  | PID0     | Set the PID value that needs to be filtered. | RW |

### 27.3.10 TSSI DMA Descriptor Address (TSSDA)

| TSSDA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x0060 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|
| Bit   | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16     | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |
|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |
| RST   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |

| Bits   | Name | Description                                                                                                                                                                                                                             | RW |
|--------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| [31:0] | DA   | Descriptor Address.<br>It must be aligned to 4-word.<br>It can be written by software and copied from the descriptor. Software should write the descriptor address before enable the DMA, and only the first DA needed if LINK enabled. | RW |

### 27.3.11 TSSI DMA Target Address (TSDTA)

| TDTA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x0064 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit  | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16     | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|      | TA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name | Description                                                                                   | R |
|------|------|-----------------------------------------------------------------------------------------------|---|
| 31:0 | TA   | Target Address.<br>It should be aligned to 32-word.<br>It will be copied from the descriptor. | R |

### 27.3.12 TSSI DMA Identifier (TSDID)

| TDID |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x0068 |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16     | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|      | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        | DID |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                        | R |
|-------|----------|----------------------------------------------------------------------------------------------------|---|
| 31:16 | Reserved | Writing has no effect, read as zero.                                                               | R |
| 15:0  | DID      | Descriptor Identifier.<br>It will be copied from the descriptor.<br>It will be copied to TDST.DID. | R |

### 27.3.13 TSSI DMA Command (TSDCMD)

| TDCMD |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x006C |          |      |     |      |      |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
|-------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----------|------|-----|------|------|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|
| Bit   | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16     | 15       | 14   | 13  | 12   | 11   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
|       | TLEN |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        | Reserved | TEFE | TSZ | TEIE | LINK |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
| RST   | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0        | 0    | 0   | 0    | 0    | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |

| Bits | Name | Description                                                                                     | R |
|------|------|-------------------------------------------------------------------------------------------------|---|
| 31:8 | TLEN | Transfer Length.<br>Unit is word.<br>When DMA transfer end, the TLEN will be counted down to 0. | R |

|     |          |                                                                                                                                 |   |
|-----|----------|---------------------------------------------------------------------------------------------------------------------------------|---|
|     |          | This field will be copied from the descriptor.                                                                                  |   |
| 7:5 | Reserved | Writing has no effect, read as zero.                                                                                            | R |
| 4   | TEFE     | Transfer End Flag Enable.<br>0: disable TSDST.TEND<br>1: enable TSDST.TEND<br>This field will be copied from the descriptor.    | R |
| 3:2 | TSZ      | Expected Transfer Size.<br>0: 4 word<br>1: 8 word<br>2: 16 word<br>3: 32 word<br>This field will be copied from the descriptor. | R |
| 1   | TEIE     | Transfer End Interrupt Enable.<br>0, disable interrupt<br>1, enable interrupt<br>This field will be copied from the descriptor. | R |
| 0   | LINK     | Descriptor Link Enable.<br>0, disable<br>1, enable<br>This field will be copied from the descriptor.                            | R |

### 27.3.14 TSST DMA Status (TSDST)

|            |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
|------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------|
|            | <b>TDST</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x0070</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
| <b>Bit</b> | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15            | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |      |
|            | DID         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | TEND |
| <b>RST</b> | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0             | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |      |

| Bits  | Name     | Description                                                                                                    | RW |
|-------|----------|----------------------------------------------------------------------------------------------------------------|----|
| 31:16 | DID      | Copied from Descriptor Identifier when DMA interrupt occurred.<br>Write 0 will clear this field.               | RW |
| 15:1  | Reserved | Write has no effect, read as zero.                                                                             | R  |
| 0     | TEND     | Transfer End Flag.<br>0: Transfer is not finished<br>1: Transfer is finished<br>Write 0 will clear this field. | RW |

**27.3.15 TSSI Transfer Control Register (TSTC)**

|             |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |          |     |     |     |   |   |   |   |   |   |   |   |   |   |
|-------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----------|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|
| <b>TSTC</b> |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x0074</b> |    |    |          |     |     |     |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>  | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16            | 15 | 14 | 13       | 12  | 11  | 10  | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               | OP |    | Reserved | OPE | EMM | APM |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0             | 0  | 0  | 0        | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|-------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-------------------|------------------------|-------|---|-------------|---|------------------|---|------------------|---|----------|-------|---|-------------|---|------------------|---|------------------|---|----------|-------|---|-------------|---|------------------|---|------------------|---|----------|-------|---|-------------|---|------------------|---|-------------------|---|-----------|--|
| 31:2  | Reserved          | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | R  |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
| 5:4   | OP                | Optional Priority Configuration. Only used when OPE is set to 1. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>OP</th> <th>TSSI AHB Priority</th> <th>Number of Data in FIFO</th> </tr> </thead> <tbody> <tr> <td rowspan="4">2'b00</td> <td>0</td> <td><math>n \leq 16</math></td> </tr> <tr> <td>1</td> <td><math>16 &lt; n \leq 32</math></td> </tr> <tr> <td>2</td> <td><math>32 &lt; n \leq 64</math></td> </tr> <tr> <td>3</td> <td><math>64 &lt; n</math></td> </tr> <tr> <td rowspan="4">2'b01</td> <td>0</td> <td><math>n \leq 16</math></td> </tr> <tr> <td>1</td> <td><math>16 &lt; n \leq 48</math></td> </tr> <tr> <td>2</td> <td><math>48 &lt; n \leq 80</math></td> </tr> <tr> <td>3</td> <td><math>80 &lt; n</math></td> </tr> <tr> <td rowspan="4">2'b10</td> <td>0</td> <td><math>n \leq 32</math></td> </tr> <tr> <td>1</td> <td><math>32 &lt; n \leq 64</math></td> </tr> <tr> <td>2</td> <td><math>64 &lt; n \leq 96</math></td> </tr> <tr> <td>3</td> <td><math>96 &lt; n</math></td> </tr> <tr> <td rowspan="4">2'b11</td> <td>0</td> <td><math>n \leq 48</math></td> </tr> <tr> <td>1</td> <td><math>48 &lt; n \leq 80</math></td> </tr> <tr> <td>2</td> <td><math>80 &lt; n \leq 112</math></td> </tr> <tr> <td>3</td> <td><math>112 &lt; n</math></td> </tr> </tbody> </table> <p style="margin-left: 20px;">It is suggested to use 2'b10.</p> | OP | TSSI AHB Priority | Number of Data in FIFO | 2'b00 | 0 | $n \leq 16$ | 1 | $16 < n \leq 32$ | 2 | $32 < n \leq 64$ | 3 | $64 < n$ | 2'b01 | 0 | $n \leq 16$ | 1 | $16 < n \leq 48$ | 2 | $48 < n \leq 80$ | 3 | $80 < n$ | 2'b10 | 0 | $n \leq 32$ | 1 | $32 < n \leq 64$ | 2 | $64 < n \leq 96$ | 3 | $96 < n$ | 2'b11 | 0 | $n \leq 48$ | 1 | $48 < n \leq 80$ | 2 | $80 < n \leq 112$ | 3 | $112 < n$ |  |
| OP    | TSSI AHB Priority | Number of Data in FIFO                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
| 2'b00 | 0                 | $n \leq 16$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 1                 | $16 < n \leq 32$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 2                 | $32 < n \leq 64$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 3                 | $64 < n$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
| 2'b01 | 0                 | $n \leq 16$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 1                 | $16 < n \leq 48$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 2                 | $48 < n \leq 80$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 3                 | $80 < n$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
| 2'b10 | 0                 | $n \leq 32$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 1                 | $32 < n \leq 64$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 2                 | $64 < n \leq 96$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 3                 | $96 < n$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
| 2'b11 | 0                 | $n \leq 48$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 1                 | $48 < n \leq 80$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 2                 | $80 < n \leq 112$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
|       | 3                 | $112 < n$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |    |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
| 3     | Reserved          | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | R  |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
| 2     | OPE               | Optional Priority Mode Enable Control. Only used when APM is 1.<br>0: TSSI calculates the priority according to the fifo status<br>1: TSSI calculates the priority according to OP which is configured by software                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
| 1     | EME               | Emergency Mode Enable Control.<br>0: Emergency Mode Disable<br>1: Emergency Mode Enable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |
| 0     | APM               | Auto Priority Mode Enable Control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW |                   |                        |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                  |   |          |       |   |             |   |                  |   |                   |   |           |  |

|  |  |                                                                                                                                                             |  |
|--|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|  |  | 0: Auto priority mode disables. TSSI uses the priority set by arbiter<br>1: Auto priority mode enables. TSSI can use the priority according the FIFO status |  |
|--|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

## 27.4 TSSI Timing

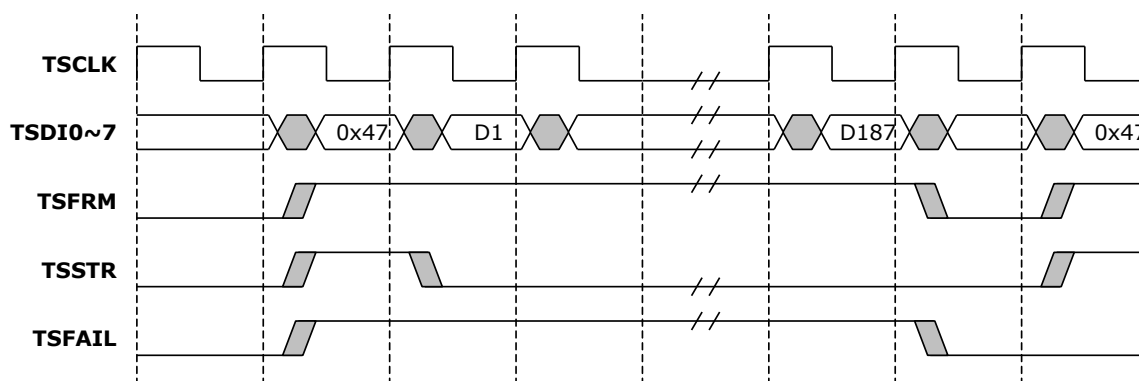


Figure 27-1 Timing waveform in parallel mode

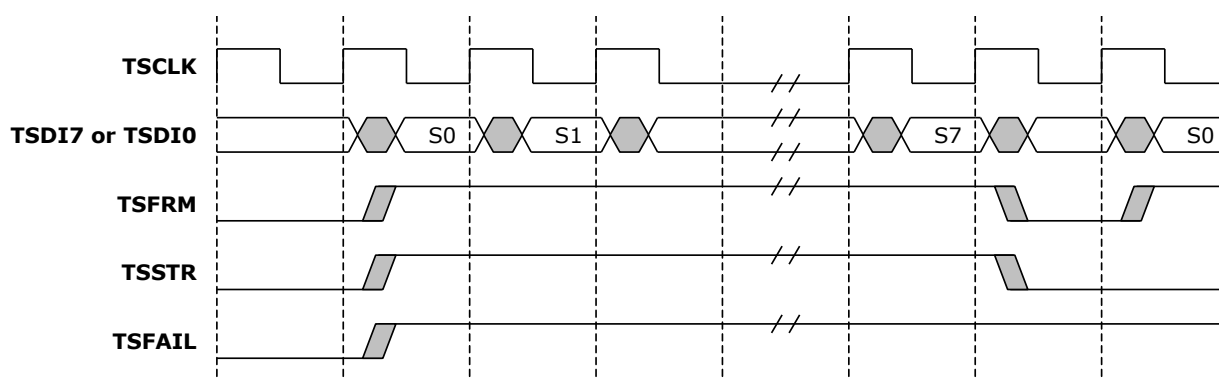


Figure 27-2 Timing waveform in serial mode

## 27.5 TSSI Guide

### 27.5.1 TSSI Operation without PID Filtering Function

- 1 Set TSCTRL to 0x03 to mask all interrupts.
- 2 Set TSCFG to choose the working mode of TSSI and the trigger value of FIFO.
- 3 Set TSENA.PID\_EN to 0 to turn off the PID filtering function.
- 4 set the DMA descriptor first if using the master DMA.
- 5 Set TSENA.DMA\_EN to 1 or 0 to decide whether to use the DMA mode or not.
- 6 Write 0x00 to TSSTAT clear all interrupt flag.
- 7 Set TSCTRL to 0x00 to enable all interrupts.
- 8 Set TSENA.ENA to 1 to turn on TSSI module.

### 27.5.2 TSSI Operation with PID Filtering Function

- 1 Set TSCTRL to 0x03 to mask all interrupts.
- 2 Set TSCFG to choose the working mode of TSSI and the trigger value of FIFO.
- 3 Set TSENA.PID\_EN to 1 to turn on the PID filtering function.
- 4 set the DMA descriptor first if using the master DMA.
- 5 Set TSENA.DMA\_EN to 1 or 0 to decide whether to use the DMA mode or not.
- 6 Write 0x00 to TSSTAT clear all interrupt flag.
- 7 Set TSCTRL to 0x00 to enable all interrupts.
- 8 Set TSENA.ENA to 1 to turn on TSSI module.
- 9 Change TSPID registers and then set TSPID to enable the PID filter.
- 10 When PID in TS package is equal to the value in TSPID register, the TS package will be getting.

## 28 General-Purpose I/O Ports

### 28.1 Overview

General Purpose I/O Ports (GPIO) is used in generating and capturing application-specific input and output signals. Each port can be programmed as an output, an input or function port that serves certain peripheral. As input, pull up/down can be enabled/disabled for the port and the port also can be configured as level or edge tripped interrupt source.

#### Features:

- Each port can be configured as an input, an output or an alternate function port
- Each port can be configured as an interrupt source of low/high level or rising/falling edge triggering. Every interrupt source can be masked independently
- Each port has an internal pull-up or pull-down resistor connected. The pull-up/down resistor can be disabled
- GPIO output 6 interrupts, 1 for every group, to INTC

#### 28.1.1 GPIO Port A Summary

Table 28-1 GPIO Port A summary

| Bit N | PA N | Pull (U/D)  | Shared Function Port Selected by |               |             |   | Note |
|-------|------|-------------|----------------------------------|---------------|-------------|---|------|
|       |      |             | 0                                | 1             | 2           | 3 |      |
| 0     | 00   | U           | sd0(io)                          | -             | -           | - |      |
| 1     | 01   | U           | sd1(io)                          | -             | -           | - |      |
| 2     | 02   | U           | sd2(io)                          | -             | -           | - |      |
| 3     | 03   | U           | sd3(io)                          | -             | -           | - |      |
| 4     | 04   | U           | sd4(io)                          | mssc0_d4(io)  | -           | - |      |
| 5     | 05   | U           | sd5(io)                          | mssc0_d5(io)  | -           | - |      |
| 6     | 06   | U           | sd6(io)                          | mssc0_d6(io)  | -           | - |      |
| 7     | 07   | U           | sd7(io)                          | mssc0_d7(io)  | -           | - |      |
| 16    | 16   | U<br>rst-pe | rd_(o)                           | -             | -           | - |      |
| 17    | 17   | U<br>rst-pe | we_(o)                           | -             | -           | - |      |
| 18    | 18   | U<br>rst-pe | fre_(o)                          | mssc0_clk(o)  | ssi0_clk(o) | - |      |
| 19    | 19   | U<br>rst-pe | fwe_(o)                          | mssc0_cmd(io) | ssi0_dt(o)  | - |      |



|    |    |             |          |              |              |   |      |
|----|----|-------------|----------|--------------|--------------|---|------|
| 20 | 20 | U           | -        | msc0_d0(io)  | ssi0_dr(i)   | - | 1    |
| 21 | 21 | U<br>rst-pe | cs1_(o)  | msc0_d1(io)  | ssi0_dt(o)   | - |      |
| 22 | 22 | U<br>rst-pe | cs2_(o)  | msc0_d2(io)  | -            | - |      |
| 23 | 23 | U<br>rst-pe | cs3_(o)  | msc0_d3(io)  | ssi0_ce0_(o) | - |      |
| 24 | 24 | U<br>rst-pe | cs4_(o)  | msc0_rst_(o) | -            | - |      |
| 25 | 25 | U<br>rst-pe | cs5_(o)  | -            | ssi0_ce0_(o) | - |      |
| 26 | 26 | U<br>rst-pe | cs6_(o)  | rdwr_(o)     | -            | - |      |
| 27 | 27 | U           | wait_(i) | -            | ssi0_dr(i)   | - |      |
| 28 | 28 | U           | -        | -            | ssi0_dt(o)   | - |      |
| 29 | 29 | U           | ndqs(io) | -            | -            | - |      |
| 30 | 30 | -           | -        | -            | -            | - | 6    |
| 31 | 31 | -           | -        | -            | -            | - | 7,11 |

### 28.1.2 GPIO Port B Summary

Table 28-2 GPIO Port B summary

| Bit N | PB N | Pull (U/D)  | Shared Function Port Selected by |             |   |   | Note |
|-------|------|-------------|----------------------------------|-------------|---|---|------|
|       |      |             | 0                                | 1           | 2 | 3 |      |
| 0     | 00   | D<br>rst-pe | sa0_cl(o)                        | -           | - | - | 8    |
| 1     | 01   | D<br>rst-pe | sa1_al(o)                        | -           | - | - | 9    |
| 2     | 02   | U           | sa2(o)                           | -           | - | - |      |
| 3     | 03   | U           | sa3(o)                           | -           | - | - |      |
| 4     | 04   | U           | sa4(o)                           | -           | - | - |      |
| 5     | 05   | U           | sa5(o)                           | ssi0_clk(o) | - | - |      |
| 6     | 06   | U           | cim_pclk(i)                      | -           | - | - |      |
| 7     | 07   | U           | cim_hsyn(i)                      | -           | - | - |      |
| 8     | 08   | U           | cim_vsyn(i)                      | -           | - | - |      |
| 9     | 09   | U           | cim_mclk(o)                      | -           | - | - |      |
| 10    | 10   | D           | cim_d0(i)                        | -           | - | - |      |
| 11    | 11   | D           | cim_d1(i)                        | -           | - | - |      |
| 12    | 12   | U           | cim_d2(i)                        | -           | - | - |      |
| 13    | 13   | U           | cim_d3(i)                        | -           | - | - |      |

|    |    |             |              |              |              |           |  |
|----|----|-------------|--------------|--------------|--------------|-----------|--|
| 14 | 14 | U           | cim_d4(i)    | -            | -            | -         |  |
| 15 | 15 | U           | cim_d5(i)    | -            | -            | -         |  |
| 16 | 16 | D           | cim_d6(i)    | -            | -            | -         |  |
| 17 | 17 | D           | cim_d7(i)    | -            | -            | -         |  |
| 18 | 18 | D           | cim_d8(i)    | dmic_clk(o)  | -            | -         |  |
| 19 | 19 | D           | cim_d9(i)    | dmic_in(i)   | -            | -         |  |
| 20 | 20 | U           | msc2_d0(io)  | ssi0_dr(i)   | ssi1_dr(i)   | tsd0(i)   |  |
| 21 | 21 | U           | msc2_d1(io)  | ssi0_ce1_(o) | ssi1_ce1_(o) | tsd1(i)   |  |
| 22 | 22 | U           | tsd2(i)      | -            | -            | -         |  |
| 23 | 23 | U           | tsd3(i)      | -            | -            | -         |  |
| 24 | 24 | U           | tsd4(i)      | -            | -            | -         |  |
| 25 | 25 | U           | tsd5(i)      | -            | -            | -         |  |
| 26 | 26 | U           | tsd6(i)      | -            | -            | -         |  |
| 27 | 27 | U           | tsd7(i)      | -            | -            | -         |  |
| 28 | 28 | U           | msc2_clk(o)  | ssi0_clk(o)  | ssi1_clk(o)  | tsclk(i)  |  |
| 29 | 29 | U<br>rst-pe | msc2_cmd(io) | ssi0_dt(o)   | ssi1_dt(o)   | tsstr(i)  |  |
| 30 | 30 | U           | msc2_d2(io)  | ssi0_gpc(o)  | ssi1_gpc(o)  | tsfail(i) |  |
| 31 | 31 | U<br>rst-pe | msc2_d3(io)  | ssi0_ce0_(o) | ssi1_ce0_(o) | tsfrm(i)  |  |

### 28.1.3 GPIO Port C Summary

Table 28-3 GPIO Port C summary

| Bit<br>N | PC<br>N | Pull<br>(U/D) | Shared Function Port Selected by |            |              |   | Note |
|----------|---------|---------------|----------------------------------|------------|--------------|---|------|
|          |         |               | 0                                | 1          | 2            | 3 |      |
| 0        | 00      | U             | lcd_b0(o)                        | lcd_rev(o) | -            | - |      |
| 1        | 01      | U             | lcd_b1(o)                        | lcd_ps(o)  | -            | - |      |
| 2        | 02      | U             | lcd_b2(o)                        | -          | -            | - |      |
| 3        | 03      | U             | lcd_b3(o)                        | -          | -            | - |      |
| 4        | 04      | U             | lcd_b4(o)                        | -          | -            | - |      |
| 5        | 05      | U             | lcd_b5(o)                        | -          | -            | - |      |
| 6        | 06      | U             | lcd_b6(o)                        | -          | -            | - |      |
| 7        | 07      | U             | lcd_b7(o)                        | -          | -            | - |      |
| 8        | 08      | U             | lcd_pclk(o)                      | -          | -            | - |      |
| 9        | 09      | U             | lcd_de(o)                        | -          | -            | - |      |
| 10       | 10      | U<br>rst-pe   | lcd_g0(o)                        | lcd_spl(o) | uart4_txd(o) | - |      |
| 11       | 11      | U             | lcd_g1(o)                        | -          | -            | - |      |
| 12       | 12      | U             | lcd_g2(o)                        | -          | -            | - |      |

|    |    |   |              |            |              |   |  |
|----|----|---|--------------|------------|--------------|---|--|
| 13 | 13 | U | lcd_g3(o)    | -          | -            | - |  |
| 14 | 14 | U | lcd_g4(o)    | -          | -            | - |  |
| 15 | 15 | U | lcd_g5(o)    | -          | -            | - |  |
| 16 | 16 | U | lcd_g6(o)    | -          | -            | - |  |
| 17 | 17 | U | lcd_g7(o)    | -          | -            | - |  |
| 18 | 18 | U | lcd_hsyn(io) | -          | -            | - |  |
| 19 | 19 | U | lcd_vsyn(io) | -          | -            | - |  |
| 20 | 20 | U | lcd_r0(o)    | lcd_cls(o) | uart4_rxd(i) | - |  |
| 21 | 21 | U | lcd_r1(o)    | -          | -            | - |  |
| 22 | 22 | U | lcd_r2(o)    | -          | -            | - |  |
| 23 | 23 | U | lcd_r3(o)    | -          | -            | - |  |
| 24 | 24 | U | lcd_r4(o)    | -          | -            | - |  |
| 25 | 25 | U | lcd_r5(o)    | -          | -            | - |  |
| 26 | 26 | U | lcd_r6(o)    | -          | -            | - |  |
| 27 | 27 | U | lcd_r7(o)    | -          | -            | - |  |

### 28.1.4 GPIO Port D Summary

Table 28-4 GPIO Port D summary

| Bit N | PD N | Pull (U/D)  | Shared Function Port Selected by |               |   |   | Note |
|-------|------|-------------|----------------------------------|---------------|---|---|------|
|       |      |             | 0                                | 1             | 2 | 3 |      |
| 0     | 00   | U           | pcm0_do(o)                       | -             | - | - |      |
| 1     | 01   | U           | pcm0_clk(io)                     | -             | - | - |      |
| 2     | 02   | U           | pcm0_syn(io)                     | -             | - | - |      |
| 3     | 03   | U           | pcm0_di(i)                       | -             | - | - |      |
| 4     | 04   | U           | ps2_mclk(io)                     | uart2_rts_(o) | - | - |      |
| 5     | 05   | U           | ps2_mdata(io)                    | uart2_cts_(i) | - | - |      |
| 6     | 06   | U           | ps2_kclk(io)                     | uart2_rxd(i)  | - | - |      |
| 7     | 07   | U           | ps2_kdata(io)                    | uart2_txd(o)  | - | - |      |
| 8     | 08   | U           | -                                | -             | - | - |      |
| 9     | 09   | U           | -                                | clk48m(i)     | - | - |      |
| 10    | 10   | U           | pwm6(io)                         | SMB3_sda(io)  | - | - |      |
| 11    | 11   | U           | pwm7(io)                         | SMB3_sck(io)  | - | - |      |
| 12    | 12   | D           | uart3_rxd(i)                     | bclk0(io)     | - | - |      |
| 13    | 13   | D           | lrclk0(io)                       | -             | - | - |      |
| 14    | 14   | U           | -                                | -             | - | - | 10   |
| 15    | 15   | D<br>rst-pe | exclk(o)                         | -             | - | - |      |
| 16    | 16   | U           | -                                | -             | - | - |      |
| 17    | 17   | U           | -                                | -             | - | - | 2,5  |

|    |    |             |               |              |              |   |     |
|----|----|-------------|---------------|--------------|--------------|---|-----|
| 18 | 18 | U           | -             | -            | -            | - | 3,5 |
| 19 | 19 | U           | -             | -            | -            | - | 4,5 |
| 20 | 20 | U           | msc1_d0(io)   | ssi0_dr(i)   | ssi1_dr(i)   | - |     |
| 21 | 21 | U           | msc1_d1(io)   | ssi0_ce1_(o) | ssi1_ce1_(o) | - |     |
| 22 | 22 | U           | msc1_d2(io)   | ssi0_gpc(o)  | ssi1_gpc(o)  | - |     |
| 23 | 23 | U<br>rst-pe | msc1_d3(io)   | ssi0_ce0_(o) | ssi1_ce0_(o) | - |     |
| 24 | 24 | U           | msc1_clk(o)   | ssi0_clk(o)  | ssi1_clk(o)  | - |     |
| 25 | 25 | U<br>rst-pe | msc1_cmd(io)  | ssi0_dt(o)   | ssi1_dt(o)   | - |     |
| 26 | 26 | U           | uart1_rxd(i)  | -            | -            | - |     |
| 27 | 27 | U           | uart1_cts_(i) | -            | -            | - |     |
| 28 | 28 | U<br>rst-pe | uart1_txd(o)  | -            | -            | - |     |
| 29 | 29 | U<br>rst-pe | uart1_rts_(o) | -            | -            | - |     |
| 30 | 30 | U           | SMB0_sda(io)  | -            | -            | - |     |
| 31 | 31 | U           | SMB0_sck(io)  | -            | -            | - |     |

### 28.1.5 GPIO Port E Summary

Table 28-5 GPIO Port E summary

| Bit<br>N | PE<br>N | Pull<br>(U/D) | Shared Function Port Selected by |               |              |   | Note |
|----------|---------|---------------|----------------------------------|---------------|--------------|---|------|
|          |         |               | 0                                | 1             | 2            | 3 |      |
| 0        | 00      | D             | pwm0(io)                         | -             | -            | - |      |
| 1        | 01      | D             | pwm1(o)                          | -             | -            | - |      |
| 2        | 02      | U             | pwm2(o)                          | -             | -            | - |      |
| 3        | 03      | U             | pwm3(io)                         | SMB4_sda(io)  | -            | - |      |
| 4        | 04      | U             | pwm4(io)                         | SMB4_sck(io)  | -            | - |      |
| 5        | 05      | U<br>rst-pe   | pwm5(io)                         | uart3_txd(o)  | sclk_rstn(o) | - |      |
| 6        | 06      | U             | aic0_sdati(i)                    | -             | -            | - |      |
| 7        | 07      | D             | aic0_sdato(o)                    | -             | -            | - |      |
| 8        | 08      | U             | uart3_cts_(i)                    | bclk0_ad(io)  | -            | - |      |
| 9        | 09      | U<br>rst-pe   | uart3_rts_(o)                    | lrclk0_ad(io) | -            | - |      |
| 10       | 10      | D<br>rst-pe   | drvbus(o)                        | -             | -            | - |      |
| 12       | 12      | U             | -                                | SMB4_sda(io)  | -            | - |      |
| 13       | 13      | U             | -                                | SMB4_sck(io)  | -            | - |      |

|    |    |             |              |              |              |   |  |
|----|----|-------------|--------------|--------------|--------------|---|--|
| 14 | 14 | U           | ssi0_dr(i)   | ssi1_dr(i)   | -            | - |  |
| 15 | 15 | U           | ssi0_clk(o)  | ssi1_clk(o)  | -            | - |  |
| 16 | 16 | U<br>rst-pe | ssi0_ce0_(o) | ssi1_ce0_(o) | -            | - |  |
| 17 | 17 | U           | ssi0_dt(o)   | ssi1_dt(o)   | -            | - |  |
| 18 | 18 | U<br>rst-pe | ssi0_ce1_(o) | ssi1_ce1_(o) | -            | - |  |
| 19 | 19 | U           | ssi0_gpc(o)  | ssi1_gpc(o)  | -            | - |  |
| 20 | 20 | U           | msc0_d0(io)  | msc1_d0(io)  | msc2_d0(io)  | - |  |
| 21 | 21 | U           | msc0_d1(io)  | msc1_d1(io)  | msc2_d1(io)  | - |  |
| 22 | 22 | U           | msc0_d2(io)  | msc1_d2(io)  | msc2_d2(io)  | - |  |
| 23 | 23 | U           | msc0_d3(io)  | msc1_d3(io)  | msc2_d3(io)  | - |  |
| 28 | 28 | U           | msc0_clk(o)  | msc1_clk(o)  | msc2_clk(o)  | - |  |
| 29 | 29 | U           | msc0_cmd(io) | msc1_cmd(io) | msc2_cmd(io) | - |  |
| 30 | 30 | U           | SMB1_sda(io) | -            | -            | - |  |
| 31 | 31 | U           | SMB1_sck(io) | -            | -            | - |  |

### 28.1.6 GPIO Port F Summary

Table 28-6 GPIO Port F summary

| Bit N | PF N | Pull (U/D)  | Shared Function Port Selected by |             |   |   | Note |
|-------|------|-------------|----------------------------------|-------------|---|---|------|
|       |      |             | 0                                | 1           | 2 | 3 |      |
| 0     | 00   | U           | uart0_rxd(i)                     | gps_clk(i)  | - | - |      |
| 1     | 01   | U<br>rst-pe | uart0_cts_(i)                    | gps_mag(i)  | - | - |      |
| 2     | 02   | U<br>rst-pe | uart0_rts_(o)                    | gps_sig(i)  | - | - |      |
| 3     | 03   | U<br>rst-pe | uart0_txd(o)                     | -           | - | - |      |
| 4     | 04   | D           | -                                | cim_d0(i)   | - | - |      |
| 5     | 05   | D           | -                                | cim_d1(i)   | - | - |      |
| 6     | 06   | D           | -                                | cim_mclk(o) | - | - |      |
| 7     | 07   | D           | -                                | cim_pclk(i) | - | - |      |
| 8     | 08   | D           | -                                | cim_hsyn(i) | - | - |      |
| 9     | 09   | D           | -                                | cim_vsyn(i) | - | - |      |
| 10    | 10   | D           | -                                | cim_d2(i)   | - | - |      |
| 11    | 11   | D           | -                                | cim_d3(i)   | - | - |      |
| 12    | 12   | U           | -                                | cim_d4(i)   | - | - |      |
| 13    | 13   | U           | -                                | cim_d5(i)   | - | - |      |
| 14    | 14   | U           | -                                | cim_d6(i)   | - | - |      |

|    |    |   |            |              |              |   |    |
|----|----|---|------------|--------------|--------------|---|----|
| 15 | 15 | U | -          | cim_d7(i)    | -            | - |    |
| 16 | 16 | U | cim_d10(i) | -            | SMB2_sda(io) | - |    |
| 17 | 17 | U | cim_d11(i) | -            | SMB2_sck(io) | - |    |
| 18 | 18 | U | -          | -            | -            |   |    |
| 19 | 19 | D | -          | -            | -            |   |    |
| 20 | 20 | D | -          | -            | -            |   |    |
| 21 | 21 | U | -          | -            | -            |   |    |
| 22 | 22 | D | -          | -            | -            |   |    |
| 23 | 23 | U | cec(io)    |              |              |   |    |
| 24 | 24 | U | ddcscl(io) | SMB4_sck(io) |              |   |    |
| 25 | 25 | U | ddcsda(io) | SMB4_sda(io) |              |   |    |
| 26 | 26 | U | -          |              |              |   | 10 |
| 27 | 27 | U | -          |              |              |   | 10 |
| 28 | 28 | U | -          |              |              |   | 10 |
| 29 | 29 | U | -          |              |              |   | 10 |
| 30 | 30 | U | owi(io)    |              |              |   | 10 |

**NOTES:**

- 1 If NAND flash is used, this pin must be used as NAND FRB.
- 2 PD17: GPIO group D bit 17 is used as BOOT\_SEL0 input during boot.
- 3 PD18: GPIO group D bit 18 is used as BOOT\_SEL1 input during boot.
- 4 PD19: GPIO group D bit 19 is used as BOOT\_SEL2 input during boot.
- 5 BOOT\_SEL2, BOOT\_SEL1, BOOT\_SEL0 are used to select boot source and function during the processor boot.
- 6 PA30: GPIO group A bit 30 can only be used as input and interrupt, no pull-up and pull-down. It is also used to select the function between PS2 function and JTAG function of JTAG/UART3/PS2 Pins( TCK\_UART3\_RTS\_PS2\_MCLK, TMS\_UART3\_CTS\_PS2\_MDATA, TDI\_UART3\_RxD\_PS2\_KCLK and TDO\_UART3\_TxD\_PS2\_KDATA), which share the same set of pins.  
When PA30.function1 is false, select JTAG function.  
When PA30.function1 is true, select PS2 function.
- 7 PA31: GPIO group A bit 31. No corresponding pin exists for this GPIO.  
It is only used to select the function between UART and JTAG of JTAG/UART3/PS2 Pins (TCK\_UART3\_RTS\_PS2\_MCLK,TMS\_UART3\_CTS\_PS2\_MDATA,TDI\_UART3\_RxD\_PS2\_KCLK and TDO\_UART3\_TxD\_PS2\_KDATA), which share the same set of pins, by using register PASEL [31].  
When PA31.function1 is false, select JTAG function.  
When PA31.function1 is true, select UART function.
- 8 If NAND flash is used, this pin must be used as NAND CLE.
- 9 If NAND flash is used, this pin must be used as NAND ALE.
- 10 PD14,PF26-30: this pin is generally used from RTC. If this pin is intended to be used as GPIO or interrupt, please refer to RTC spec to configure the corresponding registers.
- 11 The pull enable of PA31 is used to control UART and JTAG of JTAG/UART3/PS2 Pins.

## 28.2 Registers Description

Table 28-7 summarized all memory-mapped registers, which can be programmed to operate GPIO port and alternate function port sharing configuration.

All registers are in 32-bits width. Usually, 1 bit in the register affects a corresponding GPIO port and every GPIO port can be operated independently.

**Table 28-7 GPIO Registers**

| Name               | Description                          | RW | Reset Value | Address    | Size |
|--------------------|--------------------------------------|----|-------------|------------|------|
| <b>GPIO PORT A</b> |                                      |    |             |            |      |
| PAPIN              | PORT A PIN Level Register            | R  | 0x????????  | 0x10010000 | 32   |
| PAINT              | PORT A Interrupt Register            | RW | 0x00000000  | 0x10010010 | 32   |
| PAINTS             | PORT A Interrupt Set Register        | W  | 0x????????  | 0x10010014 | 32   |
| PAINTC             | PORT A Interrupt Clear Register      | W  | 0x????????  | 0x10010018 | 32   |
| PAMSK              | PORT A Interrupt Mask Register       | RW | 0xFFFFFFFF  | 0x10010020 | 32   |
| PAMSKS             | PORT A Interrupt Mask Set Register   | W  | 0x????????  | 0x10010024 | 32   |
| PAMSKC             | PORT A Interrupt Mask Clear Register | W  | 0x????????  | 0x10010028 | 32   |
| PAPAT1             | PORT A Pattern 1 Register            | RW | 0xFFFFFFFF  | 0x10010030 | 32   |
| PAPAT1S            | PORT A Pattern 1 Set Register        | W  | 0x????????  | 0x10010034 | 32   |
| PAPAT1C            | PORT A Pattern 1 Clear Register      | W  | 0x????????  | 0x10010038 | 32   |
| PAPAT0             | PORT A Pattern 0 Register            | RW | 0x00000000  | 0x10010040 | 32   |
| PAPAT0S            | PORT A Pattern 0 Set Register        | W  | 0x????????  | 0x10010044 | 32   |
| PAPAT0C            | PORT A Pattern 0 Clear Register      | W  | 0x????????  | 0x10010048 | 32   |
| PAFLG              | PORT A FLAG Register                 | R  | 0x00000000  | 0x10010050 | 32   |
| PAFLGC             | PORT A FLAG Clear Register           | W  | 0x????????  | 0x10010058 | 32   |
| PAPEN              | PORT A PULL Disable Register         | RW | 0x00000000  | 0x10010070 | 32   |
| PAPENS             | PORT A PULL Disable Set Register     | W  | 0x????????  | 0x10010074 | 32   |
| PAPENC             | PORT A PULL Disable Clear Register   | W  | 0x????????  | 0x10010078 | 32   |
| PADRVL             | -                                    | RW | 0x00000000  | 0x10010080 | 32   |
| PADRVLS            | -                                    | W  | 0x????????  | 0x10010084 | 32   |
| PADRVLC            | -                                    | W  | 0x????????  | 0x10010088 | 32   |
| PADIR              | -                                    | RW | 0x00000000  | 0x10010090 | 32   |
| PADIRS             | -                                    | W  | 0x????????  | 0x10010094 | 32   |
| PADIRC             | -                                    | W  | 0x????????  | 0x10010098 | 32   |
| PADRVH             | -                                    | RW | 0x00000000  | 0x100100A0 | 32   |
| PADRVHS            | -                                    | W  | 0x????????  | 0x100100A4 | 32   |
| PADRVHC            | -                                    | W  | 0x????????  | 0x100100A8 | 32   |
|                    |                                      |    |             |            |      |
| <b>GPIO PORT B</b> |                                      |    |             |            |      |

|                    |                                      |    |            |            |    |
|--------------------|--------------------------------------|----|------------|------------|----|
| PBPIN              | PORT B PIN Level Register            | R  | 0x???????? | 0x10010100 | 32 |
| PBINT              | PORT B Interrupt Register            | RW | 0x00000000 | 0x10010110 | 32 |
| PBINTS             | PORT B Interrupt Set Register        | W  | 0x???????? | 0x10010114 | 32 |
| PBINTC             | PORT B Interrupt Clear Register      | W  | 0x???????? | 0x10010118 | 32 |
| PBMSK              | PORT B Interrupt Mask Register       | R  | 0xFFFFFFFF | 0x10010120 | 32 |
| PBMSKS             | PORT B Interrupt Mask Set Register   | W  | 0x???????? | 0x10010124 | 32 |
| PBMSKC             | PORT B Interrupt Mask Clear Register | W  | 0x???????? | 0x10010128 | 32 |
| PBPAT1             | PORT B Pattern 1 Register            | R  | 0xFFFFFFFF | 0x10010130 | 32 |
| PBPAT1S            | PORT B Pattern 1 Set Register        | W  | 0x???????? | 0x10010134 | 32 |
| PBPAT1C            | PORT B Pattern 1 Clear Register      | W  | 0x???????? | 0x10010138 | 32 |
| PBPAT0             | PORT B Pattern 0 Register            | RW | 0x00000000 | 0x10010140 | 32 |
| PBPAT0S            | PORT B Pattern 0 Set Register        | W  | 0x???????? | 0x10010144 | 32 |
| PBPAT0C            | PORT B Pattern 0 Clear Register      | W  | 0x???????? | 0x10010148 | 32 |
| PBFLG              | PORT B FLAG Register                 | R  | 0x00000000 | 0x10010150 | 32 |
| PBFLGC             | PORT B FLAG Clear Register           | W  | 0x???????? | 0x10010158 | 32 |
| PBPEN              | PORT B PULL Disable Register         | RW | 0x00000000 | 0x10010170 | 32 |
| PBPENS             | PORT B PULL Disable Set Register     | W  | 0x???????? | 0x10010174 | 32 |
| PBPENC             | PORT B PULL Disable Clear Register   | W  | 0x???????? | 0x10010178 | 32 |
| PBDRVL             | -                                    | RW | 0x00000000 | 0x10010180 | 32 |
| PBDRVLS            | -                                    | W  | 0x???????? | 0x10010184 | 32 |
| PBDRVLC            | -                                    | W  | 0x???????? | 0x10010188 | 32 |
| PBDIR              | -                                    | RW | 0x00000000 | 0x10010190 | 32 |
| PBDIRS             | -                                    | W  | 0x???????? | 0x10010194 | 32 |
| PBDIRC             | -                                    | W  | 0x???????? | 0x10010198 | 32 |
| PBDRVH             | -                                    | RW | 0x00000000 | 0x100101A0 | 32 |
| PBDRVHS            | -                                    | W  | 0x???????? | 0x100101A4 | 32 |
| PBDRVHC            | -                                    | W  | 0x???????? | 0x100101A8 | 32 |
| <b>GPIO PORT C</b> |                                      |    |            |            |    |
| PCPIN              | PORT C PIN Level Register            | R  | 0x???????? | 0x10010200 | 32 |
| PCINT              | PORT C Interrupt Register            | RW | 0x00000000 | 0x10010210 | 32 |
| PCINTS             | PORT C Interrupt Set Register        | W  | 0x???????? | 0x10010214 | 32 |
| PCINTC             | PORT C Interrupt Clear Register      | W  | 0x???????? | 0x10010218 | 32 |
| PCMSK              | PORT C Interrupt Mask Register       | R  | 0xFFFFFFFF | 0x10010220 | 32 |
| PCMSKS             | PORT C Interrupt Mask Set Register   | W  | 0x???????? | 0x10010224 | 32 |
| PCMSKC             | PORT C Interrupt Mask Clear Register | W  | 0x???????? | 0x10010228 | 32 |
| PCPAT1             | PORT C Pattern 1 Register            | R  | 0xFFFFFFFF | 0x10010230 | 32 |
| PCPAT1S            | PORT C Pattern 1 Set Register        | W  | 0x???????? | 0x10010234 | 32 |
| PCPAT1C            | PORT C Pattern 1 Clear Register      | W  | 0x???????? | 0x10010238 | 32 |
| PCPAT0             | PORT C Pattern 0 Register            | RW | 0x00000000 | 0x10010240 | 32 |
| PCPAT0S            | PORT C Pattern 0 Set Register        | W  | 0x???????? | 0x10010244 | 32 |
| PCPAT0C            | PORT C Pattern 0 Clear Register      | W  | 0x???????? | 0x10010248 | 32 |
| PCFLG              | PORT C FLAG Register                 | R  | 0x00000000 | 0x10010250 | 32 |



|                    |                                      |    |            |            |    |
|--------------------|--------------------------------------|----|------------|------------|----|
| PCFLGC             | PORT C FLAG Clear Register           | W  | 0x???????? | 0x10010258 | 32 |
| PCPEN              | PORT C PULL Disable Register         | RW | 0x00000000 | 0x10010270 | 32 |
| PCPENS             | PORT C PULL Disable Set Register     | W  | 0x???????? | 0x10010274 | 32 |
| PCPENC             | PORT C PULL Disable Clear Register   | W  | 0x???????? | 0x10010278 | 32 |
| PCDRVL             | -                                    | RW | 0x00000000 | 0x10010280 | 32 |
| PCDRVLS            | -                                    | W  | 0x???????? | 0x10010284 | 32 |
| PCDRVLC            | -                                    | W  | 0x???????? | 0x10010288 | 32 |
| PCDIR              | -                                    | RW | 0x00000000 | 0x10010290 | 32 |
| PCDIRS             | -                                    | W  | 0x???????? | 0x10010294 | 32 |
| PCDIRC             | -                                    | W  | 0x???????? | 0x10010298 | 32 |
| PCDRVH             | -                                    | RW | 0x00000000 | 0x100102A0 | 32 |
| PCDRVHS            | -                                    | W  | 0x???????? | 0x100102A4 | 32 |
| PCDRVHC            | -                                    | W  | 0x???????? | 0x100102A8 | 32 |
| <b>GPIO PORT D</b> |                                      |    |            |            |    |
| PDPIN              | PORT D PIN Level Register            | R  | 0x???????? | 0x10010300 | 32 |
| PDINT              | PORT D Interrupt Register            | RW | 0x00000000 | 0x10010310 | 32 |
| PDINTS             | PORT D Interrupt Set Register        | W  | 0x???????? | 0x10010314 | 32 |
| PDINTC             | PORT D Interrupt Clear Register      | W  | 0x???????? | 0x10010318 | 32 |
| PDMSK              | PORT D Interrupt Mask Register       | R  | 0xFFFFFFFF | 0x10010320 | 32 |
| PDMSKS             | PORT D Interrupt Mask Set Register   | W  | 0x???????? | 0x10010324 | 32 |
| PDMSKC             | PORT D Interrupt Mask Clear Register | W  | 0x???????? | 0x10010328 | 32 |
| PDPAT1             | PORT D Pattern 1 Register            | R  | 0xFFFFFFFF | 0x10010330 | 32 |
| PDPAT1S            | PORT D Pattern 1 Set Register        | W  | 0x???????? | 0x10010334 | 32 |
| PDPAT1C            | PORT D Pattern 1 Clear Register      | W  | 0x???????? | 0x10010338 | 32 |
| PDPAT0             | PORT D Pattern 0 Register            | RW | 0x00000000 | 0x10010340 | 32 |
| PDPAT0S            | PORT D Pattern 0 Set Register        | W  | 0x???????? | 0x10010344 | 32 |
| PDPAT0C            | PORT D Pattern 0 Clear Register      | W  | 0x???????? | 0x10010348 | 32 |
| PDFLG              | PORT D FLAG Register                 | R  | 0x00000000 | 0x10010350 | 32 |
| PDFLGC             | PORT D FLAG Clear Register           | W  | 0x???????? | 0x10010358 | 32 |
| PDPEN              | PORT D PULL Disable Register         | RW | 0x00000000 | 0x10010370 | 32 |
| PDPENS             | PORT D PULL Disable Set Register     | W  | 0x???????? | 0x10010374 | 32 |
| PDPENC             | PORT D PULL Disable Clear Register   | W  | 0x???????? | 0x10010378 | 32 |
| PDDRVL             | -                                    | RW | 0x00000000 | 0x10010380 | 32 |
| PDDRVLS            | -                                    | W  | 0x???????? | 0x10010384 | 32 |
| PDDRVLC            | -                                    | W  | 0x???????? | 0x10010388 | 32 |
| PDDIR              | -                                    | RW | 0x00000000 | 0x10010390 | 32 |
| PDDIRS             | -                                    | W  | 0x???????? | 0x10010394 | 32 |
| PDDIRC             | -                                    | W  | 0x???????? | 0x10010398 | 32 |
| PDDRVH             | -                                    | RW | 0x00000000 | 0x100103A0 | 32 |
| PDDRVHS            | -                                    | W  | 0x???????? | 0x100103A4 | 32 |
| PDDRVHC            | -                                    | W  | 0x???????? | 0x100103A8 | 32 |

| GPIO PORT E |                                      |    |            |            |    |
|-------------|--------------------------------------|----|------------|------------|----|
| PEPIN       | PORT E PIN Level Register            | R  | 0x???????? | 0x10010400 | 32 |
| PEINT       | PORT E Interrupt Register            | RW | 0x00000000 | 0x10010410 | 32 |
| PEINTS      | PORT E Interrupt Set Register        | W  | 0x???????? | 0x10010414 | 32 |
| PEINTC      | PORT E Interrupt Clear Register      | W  | 0x???????? | 0x10010418 | 32 |
| PEMSK       | PORT E Interrupt Mask Register       | R  | 0xFFFFFFFF | 0x10010420 | 32 |
| PEMSKS      | PORT E Interrupt Mask Set Register   | W  | 0x???????? | 0x10010424 | 32 |
| PEMSKC      | PORT E Interrupt Mask Clear Register | W  | 0x???????? | 0x10010428 | 32 |
| PEPAT1      | PORT E Pattern 1 Register            | R  | 0xFFFFFFFF | 0x10010430 | 32 |
| PEPAT1S     | PORT E Pattern 1 Set Register        | W  | 0x???????? | 0x10010434 | 32 |
| PEPAT1C     | PORT E Pattern 1 Clear Register      | W  | 0x???????? | 0x10010438 | 32 |
| PEPAT0      | PORT E Pattern 0 Register            | RW | 0x00000000 | 0x10010440 | 32 |
| PEPAT0S     | PORT E Pattern 0 Set Register        | W  | 0x???????? | 0x10010444 | 32 |
| PEPAT0C     | PORT E Pattern 0 Clear Register      | W  | 0x???????? | 0x10010448 | 32 |
| PEFLG       | PORT E FLAG Register                 | R  | 0x00000000 | 0x10010450 | 32 |
| PEFLGC      | PORT E FLAG Clear Register           | W  | 0x???????? | 0x10010458 | 32 |
| PEPEN       | PORT E PULL Disable Register         | RW | 0x00000000 | 0x10010470 | 32 |
| PEPENS      | PORT E PULL Disable Set Register     | W  | 0x???????? | 0x10010474 | 32 |
| PEPENC      | PORT E PULL Disable Clear Register   | W  | 0x???????? | 0x10010478 | 32 |
| PEDRVL      | -                                    | RW | 0x00000000 | 0x10010480 | 32 |
| PEDRVLS     | -                                    | W  | 0x???????? | 0x10010484 | 32 |
| PEDRVLC     | -                                    | W  | 0x???????? | 0x10010488 | 32 |
| PEDIR       | -                                    | RW | 0x00000000 | 0x10010490 | 32 |
| PEDIRS      | -                                    | W  | 0x???????? | 0x10010494 | 32 |
| PEDIRC      | -                                    | W  | 0x???????? | 0x10010498 | 32 |
| PEDRVH      | -                                    | RW | 0x00000000 | 0x100104A0 | 32 |
| PEDRVHS     | -                                    | W  | 0x???????? | 0x100104A4 | 32 |
| PEDRVHC     | -                                    | W  | 0x???????? | 0x100104A8 | 32 |
| GPIO PORT F |                                      |    |            |            |    |
| PFPIN       | PORT F PIN Level Register            | R  | 0x???????? | 0x10010500 | 32 |
| PFINT       | PORT F Interrupt Register            | RW | 0x00000000 | 0x10010510 | 32 |
| PFINTS      | PORT F Interrupt Set Register        | W  | 0x???????? | 0x10010514 | 32 |
| PFINTC      | PORT F Interrupt Clear Register      | W  | 0x???????? | 0x10010518 | 32 |
| PFMSK       | PORT F Interrupt Mask Register       | R  | 0xFFFFFFFF | 0x10010520 | 32 |
| PFMSKS      | PORT F Interrupt Mask Set Register   | W  | 0x???????? | 0x10010524 | 32 |
| PFMSKC      | PORT F Interrupt Mask Clear Register | W  | 0x???????? | 0x10010528 | 32 |
| PFPAT1      | PORT F Pattern 1 Register            | R  | 0xFFFFFFFF | 0x10010530 | 32 |
| PFPAT1S     | PORT F Pattern 1 Set Register        | W  | 0x???????? | 0x10010534 | 32 |
| PFPAT1C     | PORT F Pattern 1 Clear Register      | W  | 0x???????? | 0x10010538 | 32 |
| PFPAT0      | PORT F Pattern 0 Register            | RW | 0x00000000 | 0x10010540 | 32 |
| PFPAT0S     | PORT F Pattern 0 Set Register        | W  | 0x???????? | 0x10010544 | 32 |
| PFPAT0C     | PORT F Pattern 0 Clear Register      | W  | 0x???????? | 0x10010548 | 32 |

|         |                                    |    |            |            |    |
|---------|------------------------------------|----|------------|------------|----|
| PFFLG   | PORT F FLAG Register               | R  | 0x00000000 | 0x10010550 | 32 |
| PFFLGC  | PORT F FLAG Clear Register         | W  | 0x???????? | 0x10010558 | 32 |
| PFPEN   | PORT F PULL Disable Register       | RW | 0x00000000 | 0x10010570 | 32 |
| PFPENS  | PORT F PULL Disable Set Register   | W  | 0x???????? | 0x10010574 | 32 |
| PFPENC  | PORT F PULL Disable Clear Register | W  | 0x???????? | 0x10010578 | 32 |
| PFDRVL  | -                                  | RW | 0x00000000 | 0x10010580 | 32 |
| PFDRVLS | -                                  | W  | 0x???????? | 0x10010584 | 32 |
| PFDRVLC | -                                  | W  | 0x???????? | 0x10010588 | 32 |
| PFDIR   | -                                  | RW | 0x00000000 | 0x10010590 | 32 |
| PFDIRS  | -                                  | W  | 0x???????? | 0x10010594 | 32 |
| PFDIRC  | -                                  | W  | 0x???????? | 0x10010598 | 32 |
| PFDRVH  | -                                  | RW | 0x00000000 | 0x100105A0 | 32 |
| PFDRVHS | -                                  | W  | 0x???????? | 0x100105A4 | 32 |
| PFDRVHC | -                                  | W  | 0x???????? | 0x100105A8 | 32 |

**NOTE:** PX\*\*\*\* in the description of register as follows means PA\*\*\*\*, PB\*\*\*\*, PC\*\*\*\*, PD\*\*\*\*, PE\*\*\*\* and PF\*\*\*\*.

### 28.2.1 PORT PIN Level Registers (PxPIN)

PAPIN, PBPIN, PCPIN, PDPIN, PEPIN and PFPIN are six 32-bit PORT PIN level registers. They are read-only registers.

|     |                                             |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |                                                                           |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|-----|---------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|     | PAPIN, PBPIN, PCPIN,<br>PDPIN, PEPIN, PFPIN |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | 0x10010000, 0x10010100, 0x10010200,<br>0x10010300, 0x10010400, 0x10010500 |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Bit | 31                                          | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15                                                                        | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|     | PINL31                                      | PINL30 | PINL29 | PINL28 | PINL27 | PINL26 | PINL25 | PINL24 | PINL23 | PINL22 | PINL21 | PINL20 | PINL19 | PINL18 | PINL17 | PINL16 | PINL15                                                                    | PINL14 | PINL13 | PINL12 | PINL11 | PINL10 | PINL09 | PINL08 | PINL07 | PINL06 | PINL05 | PINL04 | PINL03 | PINL02 | PINL01 | PINL00 |
| RST | 0                                           | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0                                                                         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

| Bits | Name   | Description                                                                                                              | R/W |
|------|--------|--------------------------------------------------------------------------------------------------------------------------|-----|
| n    | PINL n | Where n = 0 ~ 31 and PINL n = PINL0 ~ PINL31.<br>The PORT PIN level can be read by reading PINL n bit in register PXPIN. | R   |

### 28.2.2 PORT Interrupt Registers (PxINT)

PAINT, PBINT, PCINT, PDINT, PEINT and PFINT are six 32-bit interrupt registers.

|  |                                             |                                                                           |
|--|---------------------------------------------|---------------------------------------------------------------------------|
|  | PAINT, PBINT, PCINT,<br>PDINT, PEINT, PFINT | 0x10010010, 0x10010110, 0x10010210,<br>0x10010310, 0x10010410, 0x10010510 |
|--|---------------------------------------------|---------------------------------------------------------------------------|

|            |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>Bit</b> | 31    | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16    | 15    | 14    | 13    | 12    | 11    | 10    | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|            | INT31 | INT30 | INT29 | INT28 | INT27 | INT26 | INT25 | INT24 | INT23 | INT22 | INT21 | INT20 | INT19 | INT18 | INT17 | INT16 | INT15 | INT14 | INT13 | INT12 | INT11 | INT10 | INT09 | INT08 | INT07 | INT06 | INT05 | INT04 | INT03 | INT02 | INT01 | INT00 |
| <b>RST</b> | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

| Bits | Name  | Description                                                                                                                                                                      | R/W |
|------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| N    | INT n | Where n = 0 ~ 31 and INT n = INT31 ~ INT00.<br>Interrupt enable.<br>0: Corresponding pin is used as device functions or normal gpio<br>1: Corresponding pin is used as interrupt | RW  |

### 28.2.3 PORT Interrupt Set Registers (PxINTS)

PAINTS, PBINTS, PCINTS, PDINTS, PEINTS and PFINTS are six 32-bit interrupt set registers.

|            |                                                       |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |                                                                               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|------------|-------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|            | <b>PAINTS, PBINTS, PCINTS, PDINTS, PEINTS, PFINTS</b> |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | <b>0x10010014, 0x10010114, 0x10010214, 0x10010314, 0x10010414, 0x10010514</b> |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| <b>Bit</b> | 31                                                    | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15                                                                            | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|            | INTS31                                                | INTS30 | INTS29 | INTS28 | INTS27 | INTS26 | INTS25 | INTS24 | INTS23 | INTS22 | INTS21 | INTS20 | INTS19 | INTS18 | INTS17 | INTS16 | INTS15                                                                        | INTS14 | INTS13 | INTS12 | INTS11 | INTS10 | INTS09 | INTS08 | INTS07 | INTS06 | INTS05 | INTS04 | INTS03 | INTS02 | INTS01 | INTS00 |
| <b>RST</b> | ?                                                     | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?                                                                             | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      |

| Bits | Name   | Description                                                                                    | R/W |
|------|--------|------------------------------------------------------------------------------------------------|-----|
| N    | INTS n | Writing 1 to INTS n will set INT n to 1 in register PXINT.<br>Writing 0 to INTS n will no use. | W   |

### 28.2.4 PORT Interrupt Clear Registers (PxINTC)

PAINTC, PBINTC, PCINTC, PDINTC, PEINTC and PFINTC are six 32-bit interrupt clear registers.

|            |                                                       |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |                                                                               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|------------|-------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|            | <b>PAINTC, PBINTC, PCINTC, PDINTC, PEINTC, PFINTC</b> |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | <b>0x10010018, 0x10010118, 0x10010218, 0x10010318, 0x10010418, 0x10010518</b> |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| <b>Bit</b> | 31                                                    | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15                                                                            | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|            | INTC31                                                | INTC30 | INTC29 | INTC28 | INTC27 | INTC26 | INTC25 | INTC24 | INTC23 | INTC22 | INTC21 | INTC20 | INTC19 | INTC18 | INTC17 | INTC16 | INTC15                                                                        | INTC14 | INTC13 | INTC12 | INTC11 | INTC10 | INTC09 | INTC08 | INTC07 | INTC06 | INTC05 | INTC04 | INTC03 | INTC02 | INTC01 | INTC00 |
| <b>RST</b> | ?                                                     | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?                                                                             | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      |

| Bits | Name   | Description                                                | R/W |
|------|--------|------------------------------------------------------------|-----|
| n    | INTC n | Writing 1 to INTC n will set INT n to 0 in register PXINT. | W   |

|  |  |                                  |  |
|--|--|----------------------------------|--|
|  |  | Writing 0 to INTC n will no use. |  |
|--|--|----------------------------------|--|

### 28.2.5 PORT Mask Registers (PxMSK)

PAMSK, PBMSK, PCMSK, PDMSK, PEMSK and PFMSK are six 32-bit PORT MASK registers.

|     |  | PAMSK, PBMSK, PCMSK,<br>PDMSK, PEMSK, PFMSK |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       | 0x10010020, 0x10010120, 0x10010220,<br>0x10010320, 0x10010420, 0x10010520 |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-----|--|---------------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------------------------------------------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit |  | 31                                          | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16    | 15                                                                        | 14    | 13    | 12    | 11    | 10    | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|     |  | MSK31                                       | MSK30 | MSK29 | MSK28 | MSK27 | MSK26 | MSK25 | MSK24 | MSK23 | MSK22 | MSK21 | MSK20 | MSK19 | MSK18 | MSK17 | MSK16 | MSK15                                                                     | MSK14 | MSK13 | MSK12 | MSK11 | MSK10 | MSK09 | MSK08 | MSK07 | MSK06 | MSK05 | MSK04 | MSK03 | MSK02 | MSK01 | MSK00 |
| RST |  | 1                                           | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1                                                                         | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |

| Bits | Name  | Description                                                                                                                                                                                                                                                                        | R/W |
|------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| n    | MSK n | Where n = 0 ~ 31 and MSK n = MSK31 ~ MSK0.<br>When PXINT n = 1:<br>0: Enable the pin as an interrupt source<br>1: Disable the pin as an interrupt source<br>When PXINT n = 0:<br>0: Corresponding pin will be used as device function<br>1: Corresponding pin will be used as gpio | RW  |

### 28.2.6 PORT Mask Set Registers (PxMSKS)

PAMSKS, PBMSKS, PCMSKS, PDMSKS, PEMSKS and PFMSKS are six 32-bit PORT MASK set registers.

|     |  | PAMSKS, PBMSKS, PCMSKS,<br>PDMSKS, PEMSKS, PFMSKS |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | 0x10010024, 0x10010124, 0x10010224,<br>0x10010324, 0x10010424, 0x10010524 |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|-----|--|---------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit |  | 31                                                | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15                                                                        | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|     |  | MSKS31                                            | MSKS30 | MSKS29 | MSKS28 | MSKS27 | MSKS26 | MSKS25 | MSKS24 | MSKS23 | MSKS22 | MSKS21 | MSKS20 | MSKS19 | MSKS18 | MSKS17 | MSKS16 | MSKS15                                                                    | MSKS14 | MSKS13 | MSKS12 | MSKS11 | MSKS10 | MSKS09 | MSKS08 | MSKS07 | MSKS06 | MSKS05 | MSKS04 | MSKS03 | MSKS02 | MSKS01 | MSKS00 |
| RST |  | ?                                                 | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?                                                                         | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      |

| Bits | Name   | Description                                                                                    | R/W |
|------|--------|------------------------------------------------------------------------------------------------|-----|
| n    | MSKS n | Writing 1 to MSKS n will set MSK n to 1 in register PXMSK.<br>Writing 0 to MSKS n will no use. | W   |

### 28.2.7 PORT Mask Clear Registers (PxMSKC)

PAMSKC, PBMSKC, PCMSKC, PDMSKC, PEMSKC and PFMSKC are six 32-bit PORT MASK clear

registers.

|            |                                                       |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |                                                                               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|------------|-------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|            | <b>PAMSKS, PBMSKC, PCMSKC, PDMSKC, PEMSKC, PFMSKC</b> |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | <b>0x10010028, 0x10010128, 0x10010228, 0x10010328, 0x10010428, 0x10010528</b> |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| <b>Bit</b> | 31                                                    | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15                                                                            | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|            | MSKC31                                                | MSKC30 | MSKC29 | MSKC28 | MSKC27 | MSKC26 | MSKC25 | MSKC24 | MSKC23 | MSKC22 | MSKC21 | MSKC20 | MSKC19 | MSKC18 | MSKC17 | MSKC16 | MSKC15                                                                        | MSKC14 | MSKC13 | MSKC12 | MSKC11 | MSKC10 | MSKC09 | MSKC08 | MSKC07 | MSKC06 | MSKC05 | MSKC04 | MSKC03 | MSKC02 | MSKC01 | MSKC00 |
| <b>RST</b> | ?                                                     | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?                                                                             | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      | ?      |

| Bits | Name   | Description                                                                                    | R/W |
|------|--------|------------------------------------------------------------------------------------------------|-----|
| n    | MSKC n | Writing 1 to MSKC n will set MSK n to 0 in register PXMSK.<br>Writing 0 to MSKC n will no use. | W   |

### 28.2.8 PORT PAT1/Direction Registers (PxPAT1)

PAPAT1, PBPAT1, PCPAT1, PDPAT1, PEPAT1 and PFPAT1 are six 32-bit PORT pattern1/direction registers.

|            |                                                       |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |                                                                               |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|------------|-------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|            | <b>PAPAT1, PBPAT1, PCPAT1, PDPAT1, PEPAT1, PFPAT1</b> |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | <b>0x10010030, 0x10010130, 0x10010230, 0x10010330, 0x10010430, 0x10010530</b> |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| <b>Bit</b> | 31                                                    | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15                                                                            | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|            | PAT131                                                | PAT130 | PAT129 | PAT128 | PAT127 | PAT126 | PAT125 | PAT124 | PAT123 | PAT122 | PAT121 | PAT120 | PAT119 | PAT118 | PAT117 | PAT116 | PAT115                                                                        | PAT114 | PAT113 | PAT112 | PAT111 | PAT110 | PAT109 | PAT108 | PAT107 | PAT106 | PAT105 | PAT104 | PAT103 | PAT102 | PAT101 | PAT100 |
| <b>RST</b> | 1                                                     | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1                                                                             | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      | 1      |

| Bits | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | R/W |
|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| n    | PAT1 n | Where n = 0 ~ 31 and PAT1 n = PAT131 ~ PAT10.<br>When PINT n = 1 (Interrupt function):<br>0: Level trigger interrupt<br>1: Edge trigger interrupt<br>When PINT n = 0 and PMSK = 0 (Device function):<br>0: Corresponding pin is used as device 0 or device 1 function<br>1: Corresponding pin is used as device 2 or device 3 function<br>When PINT n = 0 and PMSK = 0 (GPIO function):<br>0: Corresponding pin is used as gpio output<br>1: Corresponding pin is used as gpio input | RW  |

### 28.2.9 PORT PAT1/Direction Set Registers (PxPAT1S)

PAPAT1S, PBPAT1S, PCPAT1S, PDPAT1S, PEPAT1S and PFPAT1S are six 32-bit PORT pattern1/direction set registers.

|     |                                                         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |                                                                           |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-----|---------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------------------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|     | PAPAT1S, PBPAT1S, PCPAT1S,<br>PDPAT1S, PEPAT1S, PFPAT1S |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | 0x10010034, 0x10010134, 0x10010234,<br>0x10010334, 0x10010434, 0x10010534 |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Bit | 31                                                      | 30      | 29      | 28      | 27      | 26      | 25      | 24      | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16      | 15                                                                        | 14      | 13      | 12      | 11      | 10      | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
|     | PAT1S31                                                 | PAT1S30 | PAT1S29 | PAT1S28 | PAT1S27 | PAT1S26 | PAT1S25 | PAT1S24 | PAT1S23 | PAT1S22 | PAT1S21 | PAT1S20 | PAT1S19 | PAT1S18 | PAT1S17 | PAT1S16 | PAT1S15                                                                   | PAT1S14 | PAT1S13 | PAT1S12 | PAT1S11 | PAT1S10 | PAT1S09 | PAT1S08 | PAT1S07 | PAT1S06 | PAT1S05 | PAT1S04 | PAT1S03 | PAT1S02 | PAT1S01 | PAT1S00 |
| RST | ?                                                       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?                                                                         | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       |

| Bits | Name    | Description                                                                                        | R/W |
|------|---------|----------------------------------------------------------------------------------------------------|-----|
| n    | PAT1S n | Writing 1 to PAT1S n will set PAT1 n to 1 in register PXPAT1.<br>Writing 0 to PAT1S n will no use. | W   |

### 28.2.10 PORT PAT1/Direction Clear Registers (PxPAT1C)

PAPAT1C, PBPAT1C, PCPAT1C, PPAT1C, PEPAT1C and PFPAT1C are six 32-bit PORT pattern1 or direction clear registers.

|     |                                                         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |                                                                           |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-----|---------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------------------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|     | PAPAT1C, PBPAT1C, PCPAT1C,<br>PDPAT1C, PEPAT1C, PFPAT1C |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | 0x10010038, 0x10010138, 0x10010238,<br>0x10010338, 0x10010438, 0x10010538 |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Bit | 31                                                      | 30      | 29      | 28      | 27      | 26      | 25      | 24      | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16      | 15                                                                        | 14      | 13      | 12      | 11      | 10      | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
|     | PAT1C31                                                 | PAT1C30 | PAT1C29 | PAT1C28 | PAT1C27 | PAT1C26 | PAT1C25 | PAT1C24 | PAT1C23 | PAT1C22 | PAT1C21 | PAT1C20 | PAT1C19 | PAT1C18 | PAT1C17 | PAT1C16 | PAT1C15                                                                   | PAT1C14 | PAT1C13 | PAT1C12 | PAT1C11 | PAT1C10 | PAT1C09 | PAT1C08 | PAT1C07 | PAT1C06 | PAT1C05 | PAT1C04 | PAT1C03 | PAT1C02 | PAT1C01 | PAT1C00 |
| RST | ?                                                       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?                                                                         | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       |

| Bits | Name    | Description                                                                                        | R/W |
|------|---------|----------------------------------------------------------------------------------------------------|-----|
| n    | PAT1C n | Writing 1 to PAT1C n will set PAT1 n to 0 in register PXPAT1.<br>Writing 0 to PAT1C n will no use. | W   |

### 28.2.11 PORT PAT0/Data Registers (PxPAT0)

PAPAT0, PBPAT0, PCPAT0, PDPAT0, PEPAT0 and PFPAT0 are six 32-bit PORT Pattern 0 or DATA registers.

|     |                                                   |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |                                                                           |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|-----|---------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|     | PAPAT0, PBPAT0, PCPAT0,<br>PDPAT0, PEPAT0, PFPAT0 |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | 0x10010040, 0x10010140, 0x10010240,<br>0x10010340, 0x10010440, 0x10010540 |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Bit | 31                                                | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15                                                                        | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|     | PAT031                                            | PAT030 | PAT029 | PAT028 | PAT027 | PAT026 | PAT025 | PAT024 | PAT023 | PAT022 | PAT021 | PAT020 | PAT019 | PAT018 | PAT017 | PAT016 | PAT015                                                                    | PAT014 | PAT013 | PAT012 | PAT011 | PAT010 | PAT009 | PAT008 | PAT007 | PAT006 | PAT005 | PAT004 | PAT003 | PAT002 | PAT001 | PAT000 |
| RST | 0                                                 | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0                                                                         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

| Bits | Name | Description | R/W |
|------|------|-------------|-----|
|------|------|-------------|-----|

|   |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |
|---|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| N | PAT0 n | <p>Where n = 0 ~ 31 and PAT0 n = PAT00 ~ PAT031.</p> <p>When PINTn = 1 and PPAT1 = 0:</p> <p>0: Port is low level triggered interrupt input<br/>1: Port is low high triggered interrupt input</p> <p>When PINTn = 1 and PPAT1 = 1:</p> <p>0: Port is falling edge triggered interrupt input<br/>1: Port is rising edge triggered interrupt input</p> <p>When PINTn = 0 and PMSK = 0 and PPAT1 = 0:</p> <p>0: Port is pin of device 0<br/>1: Port is pin of device 1</p> <p>When PINTn = 0 and PMSK = 0 and PPAT1 = 1:</p> <p>0: Port is pin of device 2<br/>1: Port is pin of device 3</p> <p>When PINTn = 0 and PMSK = 1 and PPAT1 = 0:</p> <p>0: Port is GPIO output 0<br/>1: Port is GPIO output 1</p> | RW |
|---|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|

### 28.2.12 PORT PAT0/Data Set Registers (PxPAT0S)

PAPAT0S, PBPAT0S, PCPAT0S, PDPAT0S, PEPAT0S and PFPAT0S are six 32-bit PORT Pattern0 or DATA set registers.

|            |                                                                 |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |                                                                                   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|------------|-----------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|-----------------------------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|            | <b>PAPAT0S, PBPAT0S, PCPAT0S,<br/>PDPAT0S, PEPAT0S, PFPAT0S</b> |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | <b>0x10010044, 0x10010144, 0x10010244,<br/>0x10010344, 0x10010444, 0x10010544</b> |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| <b>Bit</b> | 31                                                              | 30      | 29      | 28      | 27      | 26      | 25      | 24      | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16      | 15                                                                                | 14      | 13      | 12      | 11      | 10      | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
|            | PAT0S31                                                         | PAT0S30 | PAT0S29 | PAT0S28 | PAT0S27 | PAT0S26 | PAT0S25 | PAT0S24 | PAT0S23 | PAT0S22 | PAT0S21 | PAT0S20 | PAT0S19 | PAT0S18 | PAT0S17 | PAT0S16 | PAT0S15                                                                           | PAT0S14 | PAT0S13 | PAT0S12 | PAT0S11 | PAT0S10 | PAT0S09 | PAT0S08 | PAT0S07 | PAT0S06 | PAT0S05 | PAT0S04 | PAT0S03 | PAT0S02 | PAT0S01 | PAT0S00 |
| <b>RST</b> | ?                                                               | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?                                                                                 | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       |

| Bits | Name    | Description                                                                                                   | R/W |
|------|---------|---------------------------------------------------------------------------------------------------------------|-----|
| n    | PAT0S n | <p>Writing 1 to PAT0S n will set PAT0 n to 1 in register PXPAT0.</p> <p>Writing 0 to PAT0S n will no use.</p> | W   |

### 28.2.13 PORT PAT0/Data Clear Registers (PxPAT0C)

PAPAT0C, PBPAT0C, PCPAT0C, PDPAT0C, PEPAT0C and PFPAT0C are six 32-bit PORT Pattern 0 or DATA clear registers.

|            |                                                                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                                                                   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------|-----------------------------------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------------------------------------------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|            | <b>PAPAT0C, PBPAT0C, PCPAT0C,<br/>PDPAT0C, PEPAT0C, PFPAT0C</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x10010048, 0x10010148, 0x10010248,<br/>0x10010348, 0x10010448, 0x10010548</b> |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b> | 31                                                              | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15                                                                                | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |



|     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-----|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|     | PAT0C31 | PAT0C30 | PAT0C29 | PAT0C28 | PAT0C27 | PAT0C26 | PAT0C25 | PAT0C24 | PAT0C23 | PAT0C22 | PAT0C21 | PAT0C20 | PAT0C19 | PAT0C18 | PAT0C17 | PAT0C16 | PAT0C15 | PAT0C14 | PAT0C13 | PAT0C12 | PAT0C11 | PAT0C10 | PAT0C09 | PAT0C08 | PAT0C07 | PAT0C06 | PAT0C05 | PAT0C04 | PAT0C03 | PAT0C02 | PAT0C01 | PAT0C00 |
| RST | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       |

| Bits | Name    | Description                                                                                        | R/W |
|------|---------|----------------------------------------------------------------------------------------------------|-----|
| n    | PAT0C n | Writing 1 to PAT0C n will set PAT0 n to 0 in register PXPAT0.<br>Writing 0 to PAT0C n will no use. | W   |

### 28.2.14 PORT FLAG Registers (PxFLG)

PAFLG, PBFLG, PCFLG, PDFLG, PEFLG and PFFLG are six 32-bit GPIO FLAG registers. They are read-only registers.

|     |                                             |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |                                                                           |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|-----|---------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|     | PAFLG, PBFLG, PCFLG,<br>PDFLG, PEFLG, PFFLG |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | 0x10010050, 0x10010150, 0x10010250,<br>0x10010350, 0x10010450, 0x10010550 |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Bit | 31                                          | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15                                                                        | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|     | FLAG31                                      | FLAG30 | FLAG29 | FLAG28 | FLAG27 | FLAG26 | FLAG25 | FLAG24 | FLAG23 | FLAG22 | FLAG21 | FLAG20 | FLAG19 | FLAG18 | FLAG17 | FLAG16 | FLAG15                                                                    | FLAG14 | FLAG13 | FLAG12 | FLAG11 | FLAG10 | FLAG09 | FLAG08 | FLAG07 | FLAG06 | FLAG05 | FLAG04 | FLAG03 | FLAG02 | FLAG01 | FLAG00 |
| RST | 0                                           | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0                                                                         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

| Bits | Name   | Description                                                                                                                                                                                                                                   | R/W |
|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| n    | FLAG n | Where n = 0 ~ 31 and FLAG n = FLAG00 ~ FLAG31.<br>FLAG n is interrupt flag bit for checking the interrupt whether to happen.<br><br>When GPIO is used as interrupt function and the interrupt happened, the FLAG n in PxFLG will be set to 1. | R   |

### 28.2.15 PORT FLAG Clear Registers (PxFLGC)

PAFLGC, PBFLGC, PCFLGC, PDFLGC, PEFLGC and PFFLGC are six 32-bit GPIO FLAG Clear registers. They are read-only registers.

|     |                                                   |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |                                                                           |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-----|---------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------------------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|     | PAFLGC, PBFLGC, PCFLGC,<br>PDFLGC, PEFLGC, PFFLGC |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | 0x10010058, 0x10010158, 0x10010258,<br>0x10010358, 0x10010458, 0x10010558 |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Bit | 31                                                | 30      | 29      | 28      | 27      | 26      | 25      | 24      | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16      | 15                                                                        | 14      | 13      | 12      | 11      | 10      | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
|     | FLAGC31                                           | FLAGC30 | FLAGC29 | FLAGC28 | FLAGC27 | FLAGC26 | FLAGC25 | FLAGC24 | FLAGC23 | FLAGC22 | FLAGC21 | FLAGC20 | FLAGC19 | FLAGC18 | FLAGC17 | FLAGC16 | FLAGC15                                                                   | FLAGC14 | FLAGC13 | FLAGC12 | FLAGC11 | FLAGC10 | FLAGC09 | FLAGC08 | FLAGC07 | FLAGC06 | FLAGC05 | FLAGC04 | FLAGC03 | FLAGC02 | FLAGC01 | FLAGC00 |
| RST | 0                                                 | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0                                                                         | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

| Bits   | Name  | Description                                                            | R/W |
|--------|-------|------------------------------------------------------------------------|-----|
| [31:0] | FLAGC | When GPIO is used as interrupt function and when write 0xffffffff, the | W   |

|  |  |                                |  |
|--|--|--------------------------------|--|
|  |  | FLAG in PXFLG will be cleared. |  |
|--|--|--------------------------------|--|

### 28.2.16 PORT PULL Disable Registers (PxPE)

PAPE, PBPE, PCPE, PDPE, PEPE and PFPE are six 32-bit PORT PULL disable registers.

|     |                                       |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |                                                                           |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |   |
|-----|---------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---|
|     | PAPE, PBPE, PCPE,<br>PDPE, PEPE, PFPE |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        | 0x10010070, 0x10010170, 0x10010270,<br>0x10010370, 0x10010470, 0x10010570 |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |   |
| Bit | 31                                    | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15                                                                        | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |   |
|     | PULL31                                | PULL30 | PULL29 | PULL28 | PULL27 | PULL26 | PULL25 | PULL24 | PULL23 | PULL22 | PULL21 | PULL20 | PULL19 | PULL18 | PULL17 | PULL16 | PULL15                                                                    | PULL14 | PULL13 | PULL12 | PULL11 | PULL10 | PULL09 | PULL08 | PULL07 | PULL06 | PULL05 | PULL04 | PULL03 | PULL02 | PULL01 | PULL00 |   |
| RST | 0                                     | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0                                                                         | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0 |

| Bits | Name   | Description                                                                                                                                                                                                                                                                            | R/W |
|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| N    | PULL n | Where n = 0 ~ 31 and PULL n = PULL0 ~ PULL31.<br>PULL n is used for setting the port to be PULL UP or PULL DOWN enable.<br>0: An internal pull up or pull down resistor connects to the port. Up or down is pin dependence<br>1: No pull up or pull down resistor connects to the port | RW  |

### 28.2.17 PORT PULL Set Registers (PxPES)

PAPES, PBPEs, PCPEs, PDPEs, PEPEs and PFPEs are six 32-bit PORT PULL set registers. They are write-only registers.

|     |                                             |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |                                                                           |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-----|---------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------------------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|     | PAPES, PBPEs, PCPEs,<br>PDPEs, PEPEs, PFPEs |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | 0x10010074, 0x10010174, 0x10010274,<br>0x10010374, 0x10010474, 0x10010574 |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Bit | 31                                          | 30      | 29      | 28      | 27      | 26      | 25      | 24      | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16      | 15                                                                        | 14      | 13      | 12      | 11      | 10      | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
|     | PULLS31                                     | PULLS30 | PULLS29 | PULLS28 | PULLS27 | PULLS26 | PULLS25 | PULLS24 | PULLS23 | PULLS22 | PULLS21 | PULLS20 | PULLS19 | PULLS18 | PULLS17 | PULLS16 | PULLS15                                                                   | PULLS14 | PULLS13 | PULLS12 | PULLS11 | PULLS10 | PULLS09 | PULLS08 | PULLS07 | PULLS06 | PULLS05 | PULLS04 | PULLS03 | PULLS02 | PULLS01 | PULLS00 |
| RST | ?                                           | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?                                                                         | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       |

| Bits | Name    | Description                                                                                      | R/W |
|------|---------|--------------------------------------------------------------------------------------------------|-----|
| n    | PULLS n | Writing 1 to PULLS n will set PULL n to 1 in register PxPE.<br>Writing 0 to PULLS n will no use. | W   |

### 28.2.18 PORT PULL Clear Registers (PxPEC)

PAPEC, PBPEC, PCPEC, PDPEC, PEPEC and PFPEC are six 32-bit PORT PULL clear registers.

|            |                                                 |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |                                                                               |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|------------|-------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|-------------------------------------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|            | <b>PAPES, PBPEC, PCPEC, PDPEC, PEPEC, PFPEC</b> |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | <b>0x10010078, 0x10010178, 0x10010278, 0x10010378, 0x10010478, 0x10010578</b> |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| <b>Bit</b> | 31                                              | 30      | 29      | 28      | 27      | 26      | 25      | 24      | 23      | 22      | 21      | 20      | 19      | 18      | 17      | 16      | 15                                                                            | 14      | 13      | 12      | 11      | 10      | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
|            | PULLC31                                         | PULLC30 | PULLC29 | PULLC28 | PULLC27 | PULLC26 | PULLC25 | PULLC24 | PULLC23 | PULLC22 | PULLC21 | PULLC20 | PULLC19 | PULLC18 | PULLC17 | PULLC16 | PULLC15                                                                       | PULLC14 | PULLC13 | PULLC12 | PULLC11 | PULLC10 | PULLC09 | PULLC08 | PULLC07 | PULLC06 | PULLC05 | PULLC04 | PULLC03 | PULLC02 | PULLC01 | PULLC00 |
| <b>RST</b> | ?                                               | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?                                                                             | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       | ?       |

| Bits | Name    | Description                                                                                      | R/W |
|------|---------|--------------------------------------------------------------------------------------------------|-----|
| n    | PULLC n | Writing 1 to PULLC n will set PULL n to 0 in register PXPE.<br>Writing 0 to PULLC n will no use. | W   |

## 28.3 Program Guide

### 28.3.1 Port Function Guide

| INT | MASK | PAT1 | PAT0 | Port Description                                                                     |
|-----|------|------|------|--------------------------------------------------------------------------------------|
| 1   | 0    | 0    | 0    | Port is low level triggered interrupt input.                                         |
| 1   | 0    | 0    | 1    | Port is high level triggered interrupt input.                                        |
| 1   | 0    | 1    | 0    | Port is fall edge triggered interrupt input.                                         |
| 1   | 0    | 1    | 1    | Port is rise edge triggered interrupt input.                                         |
| 1   | 1    | 0    | 0    | Port is low level triggered interrupt input. Interrupt is masked. Flag is recorded.  |
| 1   | 1    | 0    | 1    | Port is high level triggered interrupt input. Interrupt is masked. Flag is recorded. |
| 1   | 1    | 1    | 0    | Port is fall edge triggered interrupt input. Interrupt is masked. Flag is recorded.  |
| 1   | 1    | 1    | 1    | Port is rise edge triggered interrupt input. Interrupt is masked. Flag is recorded.  |
| 0   | 0    | 0    | 0    | Port is pin of device 0.                                                             |
| 0   | 0    | 0    | 1    | Port is pin of device 1.                                                             |
| 0   | 0    | 1    | 0    | Port is pin of device 2.                                                             |
| 0   | 0    | 1    | 1    | Port is pin of device 3.                                                             |
| 0   | 1    | 0    | 0    | Port is GPIO output 0.                                                               |
| 0   | 1    | 0    | 1    | Port is GPIO output 1.                                                               |
| 0   | 1    | 1    | ?    | Port is GPIO input.                                                                  |

## 29 SMB Controller

### 29.1 Overview

The SMB bus is a two-wire serial interface, consisting of a serial data line (SDA) and a serial clock (SCL). These wires carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate as either a “transmitter” or “receiver,” depending on the function of the device. Devices can also be considered as masters or slaves when performing data transfers. A master is the device that initializes/terminates a data transfer on the bus and generates clock signals to permit that transfer. During that time, any addressed device is considered as a slave. The SMB controller is software controlled. It behaves as a master or a slave. However, operating as a master and slave simultaneously is not supported.

#### 29.1.1 Features

- Two-wire SMB serial interface – consists of a serial data line (SDA) and a serial clock (SCL)
- Two speeds
  - Standard mode (100 Kb/s)
  - Fast mode (400 Kb/s)
- Device clock is identical with pclk
- Programmable SCL generator
- Master or slave SMB operation
- 7-bit addressing/10-bit addressing
- 16-level transmit and receive FIFOs
- Interrupt operation
- The number of devices that you can connect to the same SMB-bus is limited only by the maximum bus capacitance of 400pF
- APB interface
- 5 independent SMB channels (SMB0, SMB1, SMB2, SMB3, SMB4)

#### 29.1.2 Pin Description

Table 29-1 SMB Pin Description

| Name | Width | IO | Description      |
|------|-------|----|------------------|
| SDA  | 1-bit | IO | SMB serial data  |
| SCL  | 1-bit | IO | SMB serial clock |

Please note that dedicate pull-up resistors must be introduced on board-level. The low-to-high (rise time) transition is highly dependent on RC time constant of the bus.

Totally speaking, for standard-mode SMB-bus system, the pull up resistor depends on following parameters:

- Supply voltage
- Bus capacitance
- Number of connected devices

For fast-mode SMB-bus system, switched pull-up circuit may be essential for strict speed and load requirement. (Refer to ' Philips Semiconductor, *THE SMB-BUS SPECIFICATION*, Version 2.1. Jan, 2000')

## 29.2 Registers

### 29.2.1 Registers Memory Map

A read operation to an address location that contains unused bits results in a 0 value being returned on each of the unused bits.

Registers in SMB controller can be accessed by indicating 24-bit Address Base combined with 8-bit Address Offset.

**Table 29-2 Registers Memory Map-Address Base**

| Name | Addr Base  | Description          |
|------|------------|----------------------|
| SMB0 | 0x10050000 | Address base of SMB0 |
| SMB1 | 0x10051000 | Address base of SMB1 |
| SMB2 | 0x10052000 | Address base of SMB2 |
| SMB2 | 0x10053000 | Address base of SMB3 |
| SMB2 | 0x10054000 | Address base of SMB4 |

**Table 29-3 Registers Memory Map-Address Offset**

| Name     | Addr Offset | Description                       | Width  | RW  | Reset   |
|----------|-------------|-----------------------------------|--------|-----|---------|
| SMBCON   | 0x00        | SMB control                       | 8bits  | RW  | 0x6D    |
| SMBTAR   | 0x04        | SMB target address                | 13bits | RW  | 0x0855  |
| SMBSAR   | 0x08        | SMB slave address                 | 10bits | RW  | 0x055   |
| SMBDC    | 0x10        | SMB data buffer and command       | 9bits  | RW  | 0x000   |
| SMBSHCNT | 0x14        | Standard speed SMB SCL high count | 16bits | RW  | 0x0190  |
| SMBSLCNT | 0x18        | Standard speed SMB SCL low count  | 16bits | RW  | 0x01D6  |
| SMBFHCNT | 0x1C        | Fast speed SMB SCL high count     | 16bits | RW  | 0x003C  |
| SMBFLCNT | 0x20        | Fast speed SMB SCL low count      | 16bits | RW  | 0x0082  |
| SMBINTST | 0x2C        | SMB Interrupt Status              | 12bits | R   | 0x000   |
| SMBINTM  | 0x30        | SMB Interrupt Mask                | 12bits | R/W | 12'h8FF |
|          |             |                                   |        |     |         |
| SMBRXTL  | 0x38        | SMB RxFIFO Threshold              | 4 bits | RW  | 0xF     |
| SMBXTL   | 0x3C        | SMB TxFIFO Threshold              | 4 bits | RW  | 0xF     |
| SMBCINT  | 0x40        | Clear Interrupts                  | 1 bit  | R   | 0x0     |
| SMBCRXUF | 0x44        | Clear RXUF Interrupt              | 1 bit  | R   | 0x0     |

|                |      |                                    |         |     |        |
|----------------|------|------------------------------------|---------|-----|--------|
| SMBCRXOF       | 0x48 | Clear RX_OVER Interrupt            | 1 bit   | R   | 0x0    |
| SMBCTXOF       | 0x4C | Clear TX_OVER Interrupt            | 1 bit   | R   | 0x0    |
| SMBCRXREQ      | 0x50 | Clear RDREQ Interrupt              | 1 bit   | R   | 0x0    |
| SMBCTXABT      | 0x54 | Clear TX_ABRT Interrupt            | 1 bit   | R   | 0x0    |
| SMBCRXDN       | 0x58 | Clear RX_DONE Interrupt            | 1 bit   | R   | 0x0    |
| SMBCACT        | 0x5c | Clear ACTIVITY Interrupt           | 1 bit   | R   | 0x0    |
| SMBCSTP        | 0x60 | Clear STOP Interrupt               | 1 bit   | R   | 0x0    |
| SMBCSTT        | 0x64 | Clear START Interrupt              | 1 bit   | R   | 0x0    |
| SMBCGC         | 0x68 | Clear GEN_CALL Interrupt           | 1 bit   | R   | 0x0    |
| SMBENB         | 0x6C | SMB Enable                         | 1 bit   | RW  | 0x0    |
| SMBST          | 0x70 | SMB Status register                | 7 bits  | R   | 0x06   |
|                |      |                                    |         |     |        |
|                |      |                                    |         |     |        |
| SMBABTSRC      | 0x80 | SMB Transmit Abort Status Register | 16 bits | R   | 0x0000 |
|                |      |                                    |         |     |        |
| SMBDMACR       | 0x88 | DMA Control Register               | 2 bits  | R/W | 0x0    |
| SMBDMATDL<br>R | 0x8c | DMA Transmit Data Level            | 4 bits  | R/W | 0x0    |
| SMBDMARDL<br>R | 0x90 | DMA Receive Data Level             | 4 bits  | R/W | 0x0    |
| SMBSDASU       | 0x94 | SMB SDA Setup Register             | 8 bits  | RW  | 0x64   |
| SMBACKGC       | 0x98 | SMB ACK General Call Register      | 1 bit   | RW  | 0x1    |
| SMBENBST       | 0x9C | SMB Enable Status Register         | 3 bits  | R   | 0x0    |
| SMBSDAHD       | 0xD0 | SMB SDA Hold time Register         | 9 bits  | RW  | 0x000  |
|                |      |                                    |         |     |        |
|                |      |                                    |         |     |        |
|                |      |                                    |         |     |        |

## 29.2.2 Registers and Fields Description

### 29.2.2.1 SMBCON

SMB control register. Only STPHLD can be written during SMB disable and enable, others only can be written when SMB is disable.

| SMBCON |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x00 |    |    |    |    |    |   |   |        |        |      |      |      |     |    |   |
|--------|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|--------|--------|------|------|------|-----|----|---|
| Bit    |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5    | 4    | 3    | 2   | 1  | 0 |
|        |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    |    |    |   |   | STPHLD | SLVDIS | REST | MATP | SATP | SPD | MD |   |
| RST    |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0      | 1      | 1    | 0    | 1    | 1   | 0  | 1 |

| Bits | Name     | Description                                                                                                                                                                                                                                    | RW |
|------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 15:8 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                           | R  |
| 7    | STPHLD   | Stop Hold Enable bit.<br>0: When tx fifo is empty, a STP condition will be generated<br>1: STP will never be generated until this bit changed to 0<br>Only this bit in SMBCON can be written during SMB disable and enable status.             | RW |
| 6    | SLVDIS   | This bit controls whether SMB has its slave disabled after reset.<br>0: slave is enabled<br>1: slave is disabled                                                                                                                               | RW |
| 5    | REST     | Determines whether RESTART conditions may be sent when acting as a master.<br>0: disable<br>1: enable                                                                                                                                          | RW |
| 4    | MATP     | This bit controls whether the SMB starts its transfers in 7-bit or 10-bit addressing mode when acting as a master.<br>The function of this bit is handled by bit 12 of SMBTAR register.<br>0: 7-bit addressing<br>1: 10-bit addressing         | R  |
| 3    | SATP     | When acting as a slave, this bit controls whether the SMB responds to 7-bit or 10-bit addresses.<br>0: 7-bit addressing<br>1: 10-bit addressing                                                                                                | RW |
| 2:1  | SPD      | These bits control at which speed the SMB operates.<br>1: standard mode (100 kbps)<br>2: fast mode (400 kbps)<br><b>NOTE:</b> when these two bits are set to 2'b00 or 2'b11, the speed mode will be automatically set to 2'b10 i.e. fast mode. | RW |
| 0    | MD       | This bit controls whether the SMB master is enabled.<br>0: master disabled<br>1: master enabled                                                                                                                                                | RW |

### 29.2.2.2 SMBTAR

| SMBTAR |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x04 |    |    |      |         |             |        |   |   |   |   |   |   |   |   |   |   |
|--------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|------|---------|-------------|--------|---|---|---|---|---|---|---|---|---|---|
| Bit    |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12   | 11      | 10          | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|        |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    | MATP | SPECIAL | GC_OR_START | SMBTAR |   |   |   |   |   |   |   |   |   |   |
| RST    |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0    | 0       | 1           | 0      | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

| Bits  | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                               | R/W |
|-------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:13 | Reserved    | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                      | R   |
| 12    | MATP        | This bit controls whether the SMB starts its transfers in 7- or 10-bit addressing mode when acting as a master.<br>0: 7-bit addressing<br>1: 10-bit addressing<br><b>NOTE:</b> this bit is initially set to 0.                                                                                                                                                                                                                                                            | RW  |
| 11    | SPECIAL     | This bit indicates whether software performs a General Call or START BYTE command.<br>0: ignore the bit of GC_OR_START and use SMBTAR normally<br>1: perform special SMB command as specified in GC_OR_START bit<br><b>NOTE:</b> this bit is initially set to 1.                                                                                                                                                                                                          | RW  |
| 10    | GC_OR_START | If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START byte command is to be performed by the SMB.<br>0: General Call Address – after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the SMBINTST register. The SMB remains in General Call mode until the SPECIAL bit value (bit 11) is cleared<br>1: START BYTE                                       | RW  |
| 9:0   | SMBTAR      | This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits.<br>If the SMBTAR and SMBSAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself, but only to a slave. | RW  |

**NOTE:** It is not necessary to perform any write to this register if SMB is enabled as an SMB slave only.

### 29.2.2.3 SMBSAR

|            |               |                                       |                    |
|------------|---------------|---------------------------------------|--------------------|
|            | <b>SMBSAR</b> |                                       | <b>BASE + 0x08</b> |
| <b>Bit</b> |               | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |                    |
|            |               | Reserved                              | SMBSAR             |
| <b>RST</b> |               | 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1       |                    |

| Bits  | Name     | Description                                                                                                              | R/W |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------|-----|
| 15:10 | Reserved | Writing has no effect, read as zero.                                                                                     | R   |
| 9:0   | SMBSAR   | The SMBSAR holds the slave address when the SMB is operating as a slave. For 7-bit addressing, only SMBSAR[6:0] is used. | RW  |



|  |  |                                                                                                             |  |
|--|--|-------------------------------------------------------------------------------------------------------------|--|
|  |  | This register can be written only when the SMB interface is disabled. Writes at other times have no effect. |  |
|--|--|-------------------------------------------------------------------------------------------------------------|--|

**NOTE:** It is not necessary to perform any write to this register if SMB is enabled as an SMB master only.

### 29.2.2.4 SMBDC

SMB Rx/Tx Data Buffer and Command Register, which the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO.

| SMBDC |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x10 |    |    |    |    |    |   |     |     |   |   |   |   |   |   |   |
|-------|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|-----|-----|---|---|---|---|---|---|---|
| Bit   |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8   | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    |    |    |   | CMD | DAT |   |   |   |   |   |   |   |
| RST   |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0  | 0  | 0  | 0 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                     | R/W |
|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:9 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                            | R   |
| 8    | CMD      | This bit controls whether a read or a write is performed. This bit does not control the direction when the SMB acts as a slave. It controls only the direction when it acts as a master.<br>1: Read<br>0: Write | RW  |
| 7:0  | DAT      | This register contains the data to be transmitted or received on the SMB bus.                                                                                                                                   | RW  |

**NOTE:** this command only transfer 8-bit data combined with 1-bit CMD, extra bits on the bus will be eliminated. i.e. only 8-0 bits on the bus are accepted by SMB controller.

### 29.2.2.5 SMBSHCNT

| SMBSHCNT |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x14 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit      |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|          |  |  |  |  |  |  |  |  |  |  |  |  |  | SMBSHCNT    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST      |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bits | Name | Description | R/W |
|------|------|-------------|-----|
|------|------|-------------|-----|

|      |          |                                                                                                                                                                                                                                                                                                                                                                     |    |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 15:0 | SMBSHCNT | <p>This register must be set before any SMB bus transaction can take place to ensure proper I/O timing. The register sets the SCL clock high-period count for standard speed.</p> <p>This register can be written only when the SMB interface is disabled. Writes at other times have no effect.</p> <p>SCL high time of SMB is (SMBSHCNT + 8) SMB_clk periods.</p> | RW |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|

**NOTE:** Minimum value allowed for the SMBSHCNT registers is 6. If the set value was less than 6, it will be automatically set to 6.

### 29.2.2.6 SMBSLCNT

|            |                 |  |                                       |
|------------|-----------------|--|---------------------------------------|
|            | <b>SMBSLCNT</b> |  | <b>BASE + 0x18</b>                    |
| <b>Bit</b> |                 |  | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|            |                 |  | SMBSLCNT                              |
| <b>RST</b> |                 |  | 0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0       |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                                        | R/W |
|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:0 | SMBSLCNT | <p>This register must be set before any SMB bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed.</p> <p>This register can be written only when the SMB interface is disabled. Writes at other times have no effect.</p> <p>SCL low time of SMB is (SMBSLCNT + 1) SMB_clk periods.</p> | RW  |

**NOTE:** Minimum value that can be programmed in the SMBSLCNT registers is 8. If the set value was less than 8, it will be automatically set to 8.

### 29.2.2.7 SMBFHCNT

|            |                 |  |                                       |
|------------|-----------------|--|---------------------------------------|
|            | <b>SMBFHCNT</b> |  | <b>BASE + 0x1C</b>                    |
| <b>Bit</b> |                 |  | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|            |                 |  | SMBFHCNT                              |
| <b>RST</b> |                 |  | 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0     |

| Bits | Name     | Description                                                       | R/W |
|------|----------|-------------------------------------------------------------------|-----|
| 15:0 | SMBFHCNT | This register must be set before any SMB bus transaction can take | RW  |

|  |   |                                                                                                                                                                                                                         |  |
|--|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|  | T | place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed.<br>This register can be written only when the SMB interface is disabled.<br>Writes at other times have no effect. |  |
|--|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

**NOTE:** Minimum value allowed for the SMBSHCNT registers is 6. If the set value was less than 6, it will be automatically set to 6.

### 29.2.2.8 SMBFLCNT

|                 |  |  |  |  |  |  |  |  |  |  |  |  |  |                    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|-----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------------|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| <b>SMBFLCNT</b> |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>BASE + 0x20</b> |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>      |  |  |  |  |  |  |  |  |  |  |  |  |  | 15                 | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|                 |  |  |  |  |  |  |  |  |  |  |  |  |  |                    | SMBFLCNT |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>      |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                  | 0        | 0  | 0  | 0  | 0  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bits | Name          | Description                                                                                                                                                                                                                                                                              | R/W |
|------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:0 | SMBFLCNT<br>T | This register must be set before any SMB bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed.<br>This register can be written only when the SMB interface is disabled.<br>Writes at other times have no effect. | RW  |

**NOTE:** Minimum value that can be programmed in the SMBSLCNT registers is 8. If the set value was less than 8, it will be automatically set to 8.

### 29.2.2.9 SMBINTST

Each bit in this register has a corresponding mask bit in the SMBINTM register. These bits are cleared by reading the matching interrupt clear register.

|                 |  |  |  |  |  |  |  |  |  |  |  |  |  |                    |          |    |     |      |      |      |      |       |       |       |      |      |      |      |   |   |
|-----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------------|----------|----|-----|------|------|------|------|-------|-------|-------|------|------|------|------|---|---|
| <b>SMBINTST</b> |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>BASE + 0x2C</b> |          |    |     |      |      |      |      |       |       |       |      |      |      |      |   |   |
| <b>Bit</b>      |  |  |  |  |  |  |  |  |  |  |  |  |  | 15                 | 14       | 13 | 12  | 11   | 10   | 9    | 8    | 7     | 6     | 5     | 4    | 3    | 2    | 1    | 0 |   |
|                 |  |  |  |  |  |  |  |  |  |  |  |  |  |                    | Reserved |    | IGC | ISTT | ISTP | IACT | RXDN | TXABT | RDREQ | TXEMP | TXOF | RXFL | RXOF | RXUF |   |   |
| <b>RST</b>      |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                  | 0        | 0  | 0   | 0    | 0    | 0    | 0    | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0 | 0 |

| Bits  | Name     | Description                          | R/W |
|-------|----------|--------------------------------------|-----|
| 15:12 | Reserved | Writing has no effect, read as zero. | R   |

|    |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |   |
|----|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 11 | IGC   | Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling SMB or when the CPU reads bit 0 of the SMBCGC register. SMB stores the received data in the Rx buffer.                                                                                                                                                                                                                                                                                                                                                                                                   | R |
| 10 | ISTT  | Indicates whether a START or RESTART condition has occurred on the SMB interface regardless of whether SMB is operating in slave or master mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | R |
| 9  | ISTP  | Indicates whether a STOP condition has occurred on the SMB interface regardless of whether SMB is operating in slave or master mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | R |
| 8  | IACT  | This bit captures SMB activity and stays set until it is cleared. There are four ways to clear it: <ol style="list-style-type: none"> <li>1 Disabling the SMB</li> <li>2 Reading the SMBCACT register</li> <li>3 Reading the SMBCINT register</li> <li>4 System reset</li> </ol> Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the SMB module is idle, this bit remains set until cleared, indicating that there was activity on the bus.                                                                                                                                             | R |
| 7  | RXDN  | When the SMB is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.                                                                                                                                                                                                                                                                                                                                                                                                                 | R |
| 6  | TXABT | This bit indicates if SMB, as an SMB transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an SMB master or an SMB slave, and is referred to as a “transmit abort”. When this bit is set to 1, the SMBABTSRC register indicates the reason why the transmit abort takes places.<br><b>NOTE:</b> The SMB flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register SMBCTXABT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface. | R |
| 5  | RDREQ | This bit is set to 1 when SMB is acting as a slave and another SMB master is attempting to read data from SMB. The SMB holds the SMB bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the SMBDC register. This bit is set to 0 just after the processor reads the SMBCRREQregister.                                                                                                                                         | R |
| 4  | TXEMP | This bit is set to 1 when the transmit buffer is at or below the threshold value set in the SMB_TXTL register. It is automatically cleared by hardware when the buffer level goes above the threshold.<br><br>When the SMBENB bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1,                                                                                                                                                                                                                                                                    | R |

|   |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |   |
|---|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|   |      | provided there is activity in the master or slave state machines. When there is no longer activity, then with SMBENB = 0, this bit is set to 0.                                                                                                                                                                                                                                                                                                                                                                         |   |
| 3 | TXOF | Set during transmit if the transmit buffer is filled to SMBTX_BUFFER_DEPTH and the processor attempts to issue another SMB command by writing to the SMBDC register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when SMB_en goes to 0, this interrupt is cleared.                                                                                                                                                                                 | R |
| 2 | RXFL | Set when the receive buffer reaches or goes above the SMB_RXTL threshold in the SMB_RXTL register. It is automatically cleared by hardware when buffer level goes below the threshold.<br><br>It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (SMBENB[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the SMBENB bit 0 is programmed with a 0, regardless of the activity that continues. | R |
| 1 | RXOF | Set if the receive buffer is completely filled to 2 and an additional byte is received from an external SMB device. The SMB acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (SMBENB[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when SMB_en goes to 0, this interrupt is cleared.                                                                                                                              | R |
| 0 | RXUF | Set if the processor attempts to read the receive buffer when it is empty by reading from the SMBDC register.<br>If the module is disabled (SMBENB[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when SMB_en goes to 0, this interrupt is cleared.                                                                                                                                                                                                                         | R |

### 29.2.2.10 SMBINTM

These bits mask their corresponding interrupt status bits. They are active low; a value of 0 prevents a bit from generating an interrupt.

| SMBINTM |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x30 |    |    |    |      |       |       |       |       |        |        |        |       |       |       |       |
|---------|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|------|-------|-------|-------|-------|--------|--------|--------|-------|-------|-------|-------|
| Bit     |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11   | 10    | 9     | 8     | 7     | 6      | 5      | 4      | 3     | 2     | 1     | 0     |
|         |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    | MIGC | MISTT | MISTP | MIACT | MRXDN | MTXABT | MRDREQ | MTXEMP | MTXOF | MRXFL | MRXOF | MRXUF |
| RST     |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0  | 1    | 0     | 0     | 0     | 1     | 1      | 1      | 1      | 1     | 1     | 1     | 1     |

| Bits  | Name     | Description                          | R/W |
|-------|----------|--------------------------------------|-----|
| 15:12 | Reserved | Writing has no effect, read as zero. | R   |

|    |        |                                                                                     |    |
|----|--------|-------------------------------------------------------------------------------------|----|
| 11 | MIGC   | These bits mask their corresponding interrupt status bits in the SMBINTST register. | RW |
| 10 | MISTT  |                                                                                     | RW |
| 9  | MISTP  |                                                                                     | RW |
| 8  | MIACT  |                                                                                     | RW |
| 7  | MRXDN  |                                                                                     | RW |
| 6  | MTXABT |                                                                                     | RW |
| 5  | MRDREQ |                                                                                     | RW |
| 4  | MTXEMP |                                                                                     | RW |
| 3  | MTXOF  |                                                                                     | RW |
| 2  | MRXFL  |                                                                                     | RW |
| 1  | MRXOF  |                                                                                     | RW |
| 0  | MRXUF  |                                                                                     | RW |

### 29.2.2.11 SMBRXTL

|            |                |                                     |                                       |
|------------|----------------|-------------------------------------|---------------------------------------|
|            | <b>SMBRXTL</b> |                                     | <b>BASE + 0x38</b>                    |
| <b>Bit</b> |                |                                     | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|            |                | Reserved                            | RXTL                                  |
| <b>RST</b> |                | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 |                                       |

| Bits | Name     | Description                                                                                                                                                   | R/W |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:5 | Reserved | Writing has no effect, read as zero.                                                                                                                          | R   |
| 3:0  | RXTL     | Receive FIFO Threshold Level.<br>Controls the level of entries that triggers the RxFIFO full interrupt.<br>A value of n sets the threshold for (n+1) entries. | RW  |

### 29.2.2.12 SMBTXTL

|            |                |                                     |                                       |
|------------|----------------|-------------------------------------|---------------------------------------|
|            | <b>SMBTXTL</b> |                                     | <b>BASE + 0x3C</b>                    |
| <b>Bit</b> |                |                                     | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
|            |                | Reserved                            | TXTL                                  |
| <b>RST</b> |                | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 |                                       |

| Bits | Name     | Description                                                                                                                                                | R/W |
|------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:5 | Reserved | Writing has no effect, read as zero.                                                                                                                       | R   |
| 3:0  | TXTL     | Transmit FIFO Threshold Level.<br>Controls the level of entries that trigger the TxFIFO empty interrupt.<br>A value of n sets the threshold for n entries. | RW  |

### 29.2.2.13 SMBCINT

| SMBCINT |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x40 |          |    |    |    |    |   |   |   |   |   |      |   |   |   |   |
|---------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----------|----|----|----|----|---|---|---|---|---|------|---|---|---|---|
| Bit     |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4    | 3 | 2 | 1 | 0 |
|         |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             | Reserved |    |    |    |    |   |   |   |   |   | CINT |   |   |   |   |
| RST     |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0        | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                                          | R/W |
|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                 | R   |
| 0    | CINT     | Read this register to clear the combined interrupt, all individual interrupts, and the SMBABTSRC register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the SMBABTSRC register for an exception to clearing SMBABTSRC. | R   |

### 29.2.2.14 SMBCRXUF

| SMBCRXUF |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x44 |          |    |    |    |    |   |   |   |   |   |       |   |   |   |   |
|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----------|----|----|----|----|---|---|---|---|---|-------|---|---|---|---|
| Bit      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4     | 3 | 2 | 1 | 0 |
|          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             | Reserved |    |    |    |    |   |   |   |   |   | CRXUF |   |   |   |   |
| RST      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0        | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                      | R/W |
|------|----------|----------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                             | R   |
| 0    | CRXUF    | Read this register to clear the RXUF interrupt (bit 0) of the SMBINTST register. | R   |

### 29.2.2.15 SMBCRXOF

| SMBCRXOF |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x48 |          |    |    |    |    |   |   |   |   |   |       |   |   |   |   |
|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----------|----|----|----|----|---|---|---|---|---|-------|---|---|---|---|
| Bit      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4     | 3 | 2 | 1 | 0 |
|          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             | Reserved |    |    |    |    |   |   |   |   |   | CRXOF |   |   |   |   |
| RST      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0        | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                   | R/W |
|------|----------|---------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                          | R   |
| 0    | CRXOF    | Read this register to clear the RXOF interrupt (bit 1) of the | R   |

|  |  |                    |  |
|--|--|--------------------|--|
|  |  | SMBINTST register. |  |
|--|--|--------------------|--|

### 29.2.2.16 SMBCTXOF

| SMBCTXOF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x4C |    |    |    |    |    |   |   |   |   |       |   |   |   |   |   |
|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|---|---|-------|---|---|---|---|---|
| Bit      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5     | 4 | 3 | 2 | 1 | 0 |
|          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    |    |    |   |   |   |   | CTXOF |   |   |   |   |   |
| RST      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |    |    |    |    |    |   |   |   |   |       |   |   |   |   |   |

| Bits | Name     | Description                                                                         | R/W |
|------|----------|-------------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                                | R   |
| 0    | CTXOF    | Read this register to clear the TX_OVER interrupt (bit 3) of the SMBINTST register. | R   |

### 29.2.2.17 SMBCRXREQ

| SMBCRXREQ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x50 |    |    |    |    |    |   |   |   |   |        |   |   |   |   |   |
|-----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|---|---|--------|---|---|---|---|---|
| Bit       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5      | 4 | 3 | 2 | 1 | 0 |
|           |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    |    |    |   |   |   |   | CLRREQ |   |   |   |   |   |
| RST       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |    |    |    |    |    |   |   |   |   |        |   |   |   |   |   |

| Bits | Name     | Description                                                                       | R/W |
|------|----------|-----------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                              | R   |
| 0    | CLRREQ   | Read this register to clear the RDREQ interrupt (bit 5) of the SMBINTST register. | R   |

### 29.2.2.18 SMBCTXABT

| SMBCTXABT |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x54 |    |    |    |    |    |   |   |   |   |        |   |   |   |   |   |
|-----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|---|---|--------|---|---|---|---|---|
| Bit       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5      | 4 | 3 | 2 | 1 | 0 |
|           |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    |    |    |   |   |   |   | CTXABT |   |   |   |   |   |
| RST       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |    |    |    |    |    |   |   |   |   |        |   |   |   |   |   |



| Bits | Name     | Description                                                                                                                                                                                                                                                                                              | R/W |
|------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                     | R   |
| 0    | CTXABT   | Read this register to clear the TX_ABRT interrupt (bit 6) of the SMBINTST register, and the SMBABTSRC register.<br>This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO.<br>Refer to Bit 9 of the SMBABTSRC register for an exception to clearing SMBABTSRC. | R   |

### 29.2.2.19 SMBCRXDN

| SMBCRXDN |          |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x58 |       |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|----------|----------|--|--|--|--|--|--|--|--|--|--|--|--|-------------|-------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| Bit      |          |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14    | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|          | Reserved |  |  |  |  |  |  |  |  |  |  |  |  |             | CRXDN |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST      |          |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0     | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                         | R/W |
|------|----------|-------------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                                | R   |
| 0    | CRXDN    | Read this register to clear the RX_DONE interrupt (bit 7) of the SMBINTST register. | R   |

### 29.2.2.20 SMBCACT

| SMBCACT |          |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x5C |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|---------|----------|--|--|--|--|--|--|--|--|--|--|--|--|-------------|------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| Bit     |          |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14   | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|         | Reserved |  |  |  |  |  |  |  |  |  |  |  |  |             | CACT |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST     |          |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0    | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                                           | R/W |
|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                  | R   |
| 0    | CACT     | Reading this register clears the ACTIVITY interrupt if the SMB is not active anymore. If the SMB module is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY | R   |

|  |  |                                             |  |
|--|--|---------------------------------------------|--|
|  |  | interrupt (bit 8) of the SMBINTST register. |  |
|--|--|---------------------------------------------|--|

### 29.2.2.21 SMBCSTP

| SMBCSTP |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x60 |    |    |    |    |    |   |   |   |   |      |   |   |   |   |   |
|---------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|---|---|------|---|---|---|---|---|
| Bit     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5    | 4 | 3 | 2 | 1 | 0 |
|         |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    |    |    |   |   |   |   | CSTP |   |   |   |   |   |
| RST     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0    | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                      | R/W |
|------|----------|----------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                             | R   |
| 0    | CSTP     | Read this register to clear the STOP interrupt (bit 9) of the SMBINTST register. | R   |

### 29.2.2.22 SMBCSTT

| SMBCSTT |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x64 |    |    |    |    |    |   |   |   |   |      |   |   |   |   |   |
|---------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|---|---|------|---|---|---|---|---|
| Bit     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5    | 4 | 3 | 2 | 1 | 0 |
|         |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    |    |    |   |   |   |   | CSTT |   |   |   |   |   |
| RST     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0    | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                        | R/W |
|------|----------|------------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                               | R   |
| 0    | CSTT     | Read this register to clear the START interrupt (bit 10) of the SMBINTST register. | R   |

### 29.2.2.23 SMBCGC

| SMBCGC |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x68 |    |    |    |    |    |   |   |   |   |     |   |   |   |   |   |
|--------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|---|---|-----|---|---|---|---|---|
| Bit    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5   | 4 | 3 | 2 | 1 | 0 |
|        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    |    |    |   |   |   |   | CGC |   |   |   |   |   |
| RST    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 |

| Bits | Name | Description | R/W |
|------|------|-------------|-----|
|      |      |             |     |

|      |          |                                                                                   |   |
|------|----------|-----------------------------------------------------------------------------------|---|
| 15:1 | Reserved | Writing has no effect, read as zero.                                              | R |
| 0    | CGC      | Read this register to clear the GEN_CALL interrupt (bit 11) of SMBINTST register. | R |

**29.2.2.24 SMBENB**

|               |  |  |  |  |  |  |  |  |  |  |  |  |  |                    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |        |
|---------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------------|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|--------|
| <b>SMBENB</b> |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>BASE + 0x6C</b> |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |        |
| <b>Bit</b>    |  |  |  |  |  |  |  |  |  |  |  |  |  | 15                 | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |        |
|               |  |  |  |  |  |  |  |  |  |  |  |  |  |                    | Reserved |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   | SMBENB |
| <b>RST</b>    |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                  | 0        | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |        |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | R/W |
|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | R   |
| 0    | SMBENB   | <p>Controls whether the SMB is enabled.</p> <p>0: Disables SMB (TX and RX FIFOs are held in an erased state)</p> <p>1: Enables SMB</p> <p>Software can disable SMB while it is active. However, it is important that care be taken to ensure that SMB is disabled properly.</p> <p>When SMB is disabled, the following occurs:</p> <ul style="list-style-type: none"> <li>- The TX FIFO and RX FIFO get flushed.</li> <li>- Status bits in the SMBINTST register are still active until SMB goes into IDLE state.</li> </ul> <p>If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the SMB stops the current transfer at the end of the current byte and does not acknowledge the transfer.</p> | R   |

**29.2.2.25 SMBST**

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.

When the SMB is disabled by writing 0 in bit 0 of the SMBENB register:

- Bits 1 and 2 are set to 1
- Bits 3 and 4 are set to 0

When the master or slave state machine goes to idle and ic\_en=0:

- Bits 5 and 6 are set to 0

|              |                    |
|--------------|--------------------|
| <b>SMBST</b> | <b>BASE + 0x70</b> |
|--------------|--------------------|

| Bit |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |        | 15     | 14  | 13   | 12  | 11   | 10  | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|-----|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------|--------|-----|------|-----|------|-----|---|---|---|---|---|---|---|---|---|---|---|
|     | Reserved |  |  |  |  |  |  |  |  |  |  |  |  |  |  | SLVACT | MSTACT | RFF | RFNE | TFE | TFNF | ACT |   |   |   |   |   |   |   |   |   |   |   |
| RST |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |        | 0      | 0   | 0    | 0   | 0    | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                               | R/W |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:7 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                      | R   |
| 6    | SLVACT   | Slave FSM Activity Status.<br>0: Slave FSM is in IDLE state<br>1: Slave FSM is not in IDLE state                                                                                                                                                                          | R   |
| 5    | MSTACT   | Master FSM Activity Status.<br>0: Master FSM is in IDLE state<br>1: Master FSM is not in IDLE state                                                                                                                                                                       | R   |
| 4    | RFF      | Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared.<br>0: Receive FIFO is not full<br>1: Receive FIFO is full                                        | R   |
| 3    | RFNE     | Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty.<br>0: Receive FIFO is empty<br>1: Receive FIFO is not empty                                                                     | R   |
| 2    | TFE      | Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt.<br>0: Transmit FIFO is not empty<br>1: Transmit FIFO is empty | R   |
| 1    | TFNF     | Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full.<br>0: Transmit FIFO is full<br>1: Transmit FIFO is not full                                                                                | R   |
| 0    | ACT      | SMB Activity Status.<br>The OR of SLVACT and MSTACT bits.                                                                                                                                                                                                                 | R   |

### 29.2.2.26 SMBTXFLR

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:

- The SMB is disabled
- There is a transmit abort—that is, TX\_ABRT bit is set in the SMB\_RAW\_INTR\_STAT register
- The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is

taken from the transmit FIFO.

|            |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>BASE+0X74</b> |       |   |   |   |   |  |
|------------|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------------|-------|---|---|---|---|--|
| <b>Bit</b> |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5                | 4     | 3 | 2 | 1 | 0 |  |
|            | Reserved |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                  | TXFLR |   |   |   |   |  |
| <b>RST</b> |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                | 0     | 0 | 0 | 0 | 0 |  |

| Bits | Name     | Description                                                    | R/W |
|------|----------|----------------------------------------------------------------|-----|
| 31:6 | Reserved | Reserved.                                                      | N/A |
| 5:0  | TXFLR    | Contains the number of valid data entries in the transmit FIFO | R   |

### 29.2.2.27 SMBRXFLR

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:

- The SMB is disabled
- Whenever there is a transmit abort caused by any of the events tracked in SMB\_ABTSRC

The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

|            |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>BASE+0X78</b> |       |   |   |   |   |  |
|------------|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------------|-------|---|---|---|---|--|
| <b>Bit</b> |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5                | 4     | 3 | 2 | 1 | 0 |  |
|            | Reserved |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                  | RXFLR |   |   |   |   |  |
| <b>RST</b> |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                | 0     | 0 | 0 | 0 | 0 |  |

| Bits | Name     | Description                                                   | R/W |
|------|----------|---------------------------------------------------------------|-----|
| 31:6 | Reserved | Reserved.                                                     | N/A |
| 5:0  | RXFLR    | Contains the number of valid data entries in the receive FIFO | R   |

### 29.2.2.28 SMBABTSRC

This register has 16 bits that indicate the source of the TX\_ABRT bit. Except for Bit 9, this register is cleared whenever the SMBCTXABT register or the SMBCINT register is read. To clear Bit 9, the source of the SBYTE\_NORSTRT must be fixed first; RESTART must be enabled (SMBCON[5]=1), the SPECIAL bit must be cleared (SMBTAR[11]), or the GC\_OR\_START bit must be cleared (SMBTAR[10]). Once the source of the SBYTE\_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the SBYTE\_NORSTRT is not fixed before

attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

| SMBABTSRC |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x80 |             |                 |          |                 |                     |               |                 |              |                |                 |              |                   |                    |                    |                    |
|-----------|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|-------------|-----------------|----------|-----------------|---------------------|---------------|-----------------|--------------|----------------|-----------------|--------------|-------------------|--------------------|--------------------|--------------------|
| Bit       |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14          | 13              | 12       | 11              | 10                  | 9             | 8               | 7            | 6              | 5               | 4            | 3                 | 2                  | 1                  | 0                  |
|           |  |  |  |  |  |  |  |  |  |  |  |  |  | SLVRD_INTX  | SLV_ARBLOST | SLVFLUSH_TXFIFO | ARB_LOST | ABRT_MASTER_DIS | ABRT_10B_RD_NORSTRT | SBYTE_NORSTRT | ABRT_HS_NORSTRT | SBYTE_ACKDET | ABRT_HS_ACKDET | ABRT_GCALL_READ | ABRT_GCALL_N | ABRT_TXDATA_NOACK | ABRT_10ADDR2_NOACK | ABRT_10ADDR1_NOACK | ABRT_7B_ADDR_NOACK |
| RST       |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0           | 0               | 0        | 0               | 0                   | 0             | 0               | 0            | 0              | 0               | 0            | 0                 | 0                  | 0                  | 0                  |

| Bits | Name                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                 | R/W |
|------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15   | SLVRD_INTX          | 1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of SMBDC register.                                                                                                                                                                                                                                                                             | R   |
| 14   | SLV_ARBLOST         | 1: Slave lost the bus while transmitting data to a remote master. SMBABTSRC[12] is set at the same time. <b>NOTE:</b> Even though the slave never “owns” the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then SMB no longer own the bus.<br>Reset value: 0x0. | R   |
| 13   | SLVFLUSH_TXFIFO     | 1: Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.<br>Reset value: 0x0.                                                                                                                                                                                                                                                                         | R   |
| 12   | ARB_LOST            | 1: Master has lost arbitration, or if SMBABTSRC[14] is also set, then the slave transmitter has lost arbitration.<br><b>NOTE:</b> SMB can be both master and slave at the same time.<br>Reset value: 0x0.                                                                                                                                                                                                                                   | R   |
| 11   | ABRT_MASTER_DIS     | 1: User tries to initiate a Master operation with the Master mode disabled.<br>Reset value: 0x0.                                                                                                                                                                                                                                                                                                                                            | R   |
| 10   | ABRT_10B_RD_NORSTRT | 1: The restart is disabled (SMBRESTART_EN bit (SMBCON[5]) = 0) and the master sends a read command in 10-bit addressing mode.<br>Reset value: 0x0.                                                                                                                                                                                                                                                                                          | R   |
| 9    | SBYTE_NOR           | To clear Bit 9, the source of the SBYTE_NORSTRT must be fixed                                                                                                                                                                                                                                                                                                                                                                               | R   |

|   |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |   |
|---|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|   | STRT               | <p>first; restart must be enabled (SMBCON[5]=1), the SPECIAL bit must be cleared (SMBTAR[11]), or the GC_OR_START bit must be cleared (SMBTAR[10]). Once the source of the SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted. 1: The restart is disabled (SMBRESTART_EN bit (SMBCON[5]) = 0) and the user is trying to send a START Byte.</p> <p>Reset value: 0x0.</p> |   |
| 8 | ABRT_HS_NORSTRT    | <p>1: The restart is disabled (SMBRESTART_EN bit (SMBCON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode.</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                                                                                           | R |
| 7 | SBYTE_ACKDET       | <p>1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                  | R |
| 6 | ABRT_HS_ACKDET     | <p>1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                      | R |
| 5 | ABRT_GCALL_READ    | <p>1: SMB in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (SMBDC[9] is set to 1).</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                                                                                | R |
| 4 | ABRT_GCALL_NOACK   | <p>1: SMB in master mode sent a General Call and no slave on the bus acknowledged the General Call.</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                       | R |
| 3 | ABRT_TXDATA_NOACK  | <p>1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledgement from the remote slave(s).</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                        | R |
| 2 | ABRT_10ADDR2_NOACK | <p>1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave.</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                                                                                                               | R |
| 1 | ABRT_10ADDR1_NOACK | <p>1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                               | R |
| 0 | ABRT_7B_ADDR_NOACK | <p>1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                          | R |

### 29.2.2.29 SMBDMACR

The register is used to enable the DMA Controller interface operation. There is a separate bit for transmit and receive. This can be programmed regardless of the state of SMB enable.

| SMBDMACR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x88 |          |    |    |    |    |   |   |   |   |   |      |      |   |   |   |  |   |   |
|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----------|----|----|----|----|---|---|---|---|---|------|------|---|---|---|--|---|---|
| Bit      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4    | 3    | 2 | 1 | 0 |  |   |   |
|          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             | Reserved |    |    |    |    |   |   |   |   |   | TDEN | RDEN |   |   |   |  |   |   |
| RST      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |          |    |    |    |    |   |   |   |   |   |      |      |   |   |   |  | 0 | 0 |

| Bits | Name     | Description                                                                                                                               | R/W |
|------|----------|-------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:2 | Reserved | Writing has no effect, read as zero.                                                                                                      | R   |
| 1    | TDEN     | <b>Transmit DMA Enable.</b><br>This bit enables/disables the transmit DMA channel.<br>0: Transmit DMA disabled<br>1: Transmit DMA enabled | R/W |
| 0    | RDEN     | <b>Receive DMA Enable.</b><br>This bit enables/disables the receive DMA channel.<br>0: Receive DMA disabled<br>1: Receive DMA enabled     | R/W |

### 29.2.2.30 SMBDMATDLR

The register is used to config dma transmit data level.

| SMBDMACR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x8c |          |    |    |    |    |   |   |   |   |   |      |   |   |   |   |  |   |   |
|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----------|----|----|----|----|---|---|---|---|---|------|---|---|---|---|--|---|---|
| Bit      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4    | 3 | 2 | 1 | 0 |  |   |   |
|          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             | Reserved |    |    |    |    |   |   |   |   |   | TDLR |   |   |   |   |  |   |   |
| RST      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |          |    |    |    |    |   |   |   |   |   |      |   |   |   |   |  | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                     | R/W |
|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:5 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                            | R   |
| 4:0  | TDLR     | <b>DMA Transmit Data Level.</b><br>This bit field controls the level at which a DMA request is made by the transmit logic. The dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value. | R/W |



### 29.2.2.31 SMBDMARDLR

The register is used to config dma receive data level.

| SMBDMACR |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x90 |          |    |    |    |    |   |   |   |   |   |      |   |   |   |   |
|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----------|----|----|----|----|---|---|---|---|---|------|---|---|---|---|
| Bit      |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4    | 3 | 2 | 1 | 0 |
|          |  |  |  |  |  |  |  |  |  |  |  |  |  |             | Reserved |    |    |    |    |   |   |   |   |   | RDLR |   |   |   |   |
| RST      |  |  |  |  |  |  |  |  |  |  |  |  |  |             |          |    |    |    |    |   |   |   |   |   |      |   |   | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                   | R/W |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:5 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                          | R   |
| 4:0  | RDLR     | <b>DMA Receive Data Level.</b><br>This bit field controls the level at which a DMA request is made by the receive logic. The dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1. | R/W |

### 29.2.2.32 SMBSDASU

This register controls the amount of time delay (in terms of number of SMB\_clk clock periods) introduced in the rising edge of SCL, relative to SDA changing, when SMB services a read request in a slave-transmitter operation. The relevant SMB requirement is t<sub>SU:DAT</sub> (NOTE 2) as detailed in the SMB Bus Specification.

**NOTE:** The length of setup time is calculated using  $[(SMBSDASU - 1) * (ic\_clk\_period)]$ , so if the user requires 10 ic\_clk periods of setup time, they should program a value of 11.

A Fast-mode SMB-bus device can be used in a Standard-mode SMB-bus system, but the requirement  $t_{SU:DAT} \geq 250$  ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line  $t_{r\_max} + t_{SU:DAT} = 1000 + 250 = 1250$  ns (according to the Standard-mode SMB-bus specification) before the SCL line is released. (Refer to ' Philips Semiconductor, *THE SMB-BUS SPECIFICATION*, Version 2.1. Jan, 2000')

| SMBSDASU |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x94 |          |    |    |    |    |   |   |       |   |   |   |   |   |   |   |   |   |
|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----------|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|---|---|
| Bit      |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |
|          |  |  |  |  |  |  |  |  |  |  |  |  |  |             | Reserved |    |    |    |    |   |   | SDASU |   |   |   |   |   |   |   |   |   |
| RST      |  |  |  |  |  |  |  |  |  |  |  |  |  |             |          |    |    |    |    |   |   |       |   |   |   |   |   | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                 | R/W |
|------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:8 | Reserved | Writing has no effect, read as zero.                                                                                                                        | R   |
| 7:0  | SDASU    | SDA Setup. It is recommended that if the required delay is 1000ns, then for an SMB_clk frequency of 10 MHz, SMBSDASU should be programmed to a value of 11. | R/W |

### 29.2.2.33 SMBACKGC

The register controls whether SMB responds with an ACK or NACK when it receives an SMB General Call address.

| SMBACKGC |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x98 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit      |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|          | Reserved |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ACKGC       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST      |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |   |

| Bits | Name     | Description                                                                                                                                                                            | R/W |
|------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:1 | Reserved | Writing has no effect, read as zero.                                                                                                                                                   | R   |
| 0    | ACKGC    | ACK General Call. When set to 1, SMB responds with an ACK (by asserting ic_data_oe) when it receives a General Call. When set to 0, the SMB does not generate General Call interrupts. | R/W |

### 29.2.2.34 SMBENBST

The register is used to report the SMB hardware status when the SMBENB register is set from 1 to 0; that is, when SMB is disabled.

If SMBENB has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.

If SMBENB has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.

| SMBENBST |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BASE + 0x9C                  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit      |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15                           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|          | Reserved |  |  |  |  |  |  |  |  |  |  |  |  |  |  | SLVRDLST<br>SLVDISB<br>SMBEN |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST      |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                            | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | R/W |
|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:3 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | R   |
| 2    | SLVRDLST | <p>Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an SMB transfer due to the setting of SMBENB from 1 to 0.</p> <p>When read as 1, SMB is deemed to have been actively engaged in an aborted SMB transfer (with matching address) and the data phase of the SMB transfer has been entered, even though a data byte has been responded with a NACK. <b>NOTE:</b> If the remote SMB master terminates the transfer with a STOP condition before the SMB has a chance to NACK a transfer, and SMBENB has been set to 0, then this bit is also set to 1. When read as 0, SMB is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer.</p> <p><b>NOTE:</b> The CPU can safely read this bit when IC_EN (bit 0) is read as 0.</p> <p>Reset value: 0x0.</p>                                                                                                                                                                                                                                                                                                                             | R   |
| 1    | SLVDISB  | <p>Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the SMBENB register from 1 to 0. This bit is set when the CPU writes a 0 to the SMBENB register while: (a) SMB is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master.</p> <p>When read as 1, SMB is deemed to have forced a NACK during any part of an SMB transfer, irrespective of whether the SMB address matches the slave address set in SMB (SMBSAR register) OR if the transfer is completed before SMBENB is set to 0 but has not taken effect.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1 If the remote SMB master terminates the transfer with a STOP condition before the SMB has a chance to NACK a transfer, and SMBENB has been set to 0, then this bit will also be set to 1. When read as 0, SMB is deemed to have been disabled when there is master activity, or when the SMB bus is idle.</li> <li>2 The CPU can safely read this bit when IC_EN (bit 0) is read as 0.</li> </ol> <p>Reset value: 0x0.</p> | R   |
| 0    | SMBEN    | <p>ic_en Status. This bit always reflects the value driven on the output port ic_en.</p> <p>When read as 1, SMB is deemed to be in an enabled state.</p> <p>When read as 0, SMB is deemed completely inactive.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | R   |

|  |  |                                                                                                                                                                       |  |
|--|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|  |  | <b>NOTE:</b> The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLVRDLST (bit 2) and SLVDISB (bit 1).<br>Reset value: 0x0. |  |
|--|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

### 29.2.2.35 SMBSDAHD

This register controls the amount of delay (in terms of SMB\_clk cycles) that introduced in the falling edge of SCL, relative to SDA changing. It works when SMB services a read request in master-transmitter operation.

The length of hold time is calculated using:  $T_{\text{delay}} = (\text{SMBSDAHD} + 1) * (\text{SMB\_clk period})$ .

|            |                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                    |    |    |    |    |    |   |       |       |   |   |   |   |   |   |   |
|------------|-----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------------|----|----|----|----|----|---|-------|-------|---|---|---|---|---|---|---|
|            | <b>SMBSDAHD</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>BASE + 0xD0</b> |    |    |    |    |    |   |       |       |   |   |   |   |   |   |   |
| <b>Bit</b> |                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15                 | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|            |                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved           |    |    |    |    |    |   | HDENB | SDAHD |   |   |   |   |   |   |   |
| <b>RST</b> |                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                  | 0  | 0  | 0  | 0  | 0  | 0 | 0     | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                                                                        | R/W |
|------|----------|------------------------------------------------------------------------------------------------------------------------------------|-----|
| 15:9 | Reserved | Reserved.                                                                                                                          | N/A |
| 8    | HDENB    | SMB Hold Enable.<br>0: The actual hold time is 1 * (SMB_clk period).<br>1: The actual hold time is (SDAHD + 1) * (SMB_clk period). | RW  |
| 7:0  | SDAHD    | SMB Hold Time.<br>In terms of number of SMB_clk clock periods                                                                      | RW  |

## 29.3 Operating Flow

This section provides information on the following topics:

- “Slave Mode Operation”
- “Master Mode Operation”
- “Disabling SMB”

**NOTE:** It is important to note that the SMB should only be set to operate as an SMB Master, or SMB Slave, but not both simultaneously. This is achieved by ensuring that bit 6 (SMBSLAVE\_DISABLE) and 0 (SMBMASTER\_MODE) of the SMBCON register are never set to 0 and 1, respectively.

### 29.3.1 SMB Behavior

The SMB can be controlled via software to be either:

- An SMB master only, communicating with other SMB slaves.

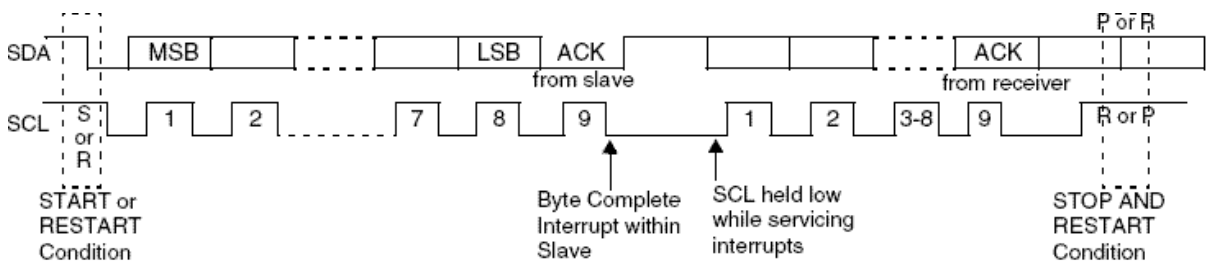
OR

- An SMB slave only, communicating with one more SMB masters.

The master is responsible for generating the clock and controlling the transfer of data. The slave is responsible for either transmitting or receiving data to/from the master. The device that is receiving data, which can be either a master or a slave, sends the acknowledgement of data. As mentioned previously, the SMB protocol also allows multiple masters to reside on the SMB bus and uses an arbitration procedure to determine bus ownership.

Each slave has a unique address that is determined by the system designer. When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave's address and a control bit (R/W) to determine if the master wants to transmit data or receive data from the slave. The slave then sends an acknowledgement (ACK) pulse after the address.

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver gets one byte of data. This transaction continues until the master terminates the transmission with a STOP condition. If the master is reading from a slave (master-receiver), the slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This transaction continues until the master terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition.



### 29.3.2 Master Mode Operation

This section includes the following topics:

- "Initial Configuration"
- "Dynamic SMBTAR or SMB10BITADDR\_MASTER Update"
- "Master Transmit and Master Receive"

#### 29.3.2.1 Configuration

To use the SMB as a master perform the following steps:

- 1 Disable the SMB by writing 0 to the SMBENB register. And wait for the SMBENBST.SMBEN = 0.
- 2 Write to the SMBCON register to set the speed mode supported (bits 2:1). Please note that

the MATP (bit4) is NOT writable. The addressing mode is controlled by Register SMBTAR.

- 3 Set the expected SCL frequency. SMBCON.SPD = 2'b01, only SMBSHCNT and SMBSLCNT are needed to be configured; SMBCON.SPD = 2'b10, only SMBFHCNT and SMBFLCNT are needed to be configured.

Supposed:

$T_{scl}$  : SMB SCL period  
 $T_{SMB\_clk}$  : SMB device clock period  
 $T_{min\_scl\_l}$ : Protocol minimum SCL low time  
 $T_{min\_scl\_h}$ : Protocol minimum SCL high time

Then can get the equation:

$$T_{scl} = T_{SMB\_clk} * ((SMB*HCNT + 8) + (SMB*LCNT + 1))$$

And the following conditions should be met:

$(SMB*HCNT + 8) * T_{SMB\_clk} \geq T_{min\_scl\_h}$   
 $(SMB*LCNT + 1) * T_{SMB\_clk} \geq T_{min\_scl\_l}$   
 $SMB*HCNT \geq 6$   
 $SMB*LCNT \geq 8$

- 4 Write to the SMBTAR register the address of the SMB device to be addressed. It also indicates whether a General Call or a START BYTE command is going to be performed by SMB. The addressing mode that master starts, either 7-bit or 10-bit addressing, is controlled by the SMB10BITADDR\_MASTER bit field (bit 12).
- 5 Enable the SMB by writing a 1 into SMBENB register. And wait for the SMBENBST.SMBEN = 1.
- 6 Now write the transfer direction and data to be sent to the SMBDC register (This can be done by DMA). If the SMBDC register is written before the SMB is enabled, the data and commands are lost as the buffers are kept cleared when SMB is not enabled.

**NOTE:** For multiple SMB transfers, perform additional writes to the TX FIFO such that the TX FIFO does not become empty during the SMB transaction. If the TX FIFO is completely emptied at any stage and SMBCON.STPHLD is 0, then further writes to the TX FIFO results in an independent SMB transaction.

### 29.3.2.2 Dynamic SMBTAR or SMB10BITADDR\_MASTER Update

The SMB supports dynamic updating of the SMBTAR (bits 9:0) and SMB10BITADDR\_MASTER (bit 12) bit fields of the SMBTAR register. You can dynamically write to the SMBTAR register provided the following conditions are met:

- 1 SMB is not enabled (SMBENB=0);
- OR
- 2 SMB is enabled (SMBENB=1);
- AND
- SMB is NOT engaged in any Master (tx, rx) operation (SMBST[5]=0);
- AND
- SMB is enabled to operate in Master mode (SMBCON[0]=1);

AND

there are NO entries in the TX FIFO (SMBST[2]=0).

### 29.3.2.3 Master Transmit and Master Receive

The SMB supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be transferred to lower byte of the SMB Rx/Tx Data Buffer and Command Register (SMBDC). The *CMD* bit (SMBDC[8]) should be written to 0 for SMB write operations.

Subsequently, a read command may be issued by writing “don’t cares” to the lower byte of the SMBDC register, and a 1 should be written to the *CMD* bit.

### 29.3.3 Slave Mode Operation

This section includes the following procedures:

- “Initial Configuration”
- “Slave-Transmitter Operation for a Single Byte”
- “Slave-Receiver Operation for a Single Byte”
- “Slave-Transfer Operation For Bulk Transfers”

#### 29.3.3.1 Initial Configuration

To use the SMB as a slave, perform the following steps:

- 1 Disable the SMB by writing a ‘0’ to bit 0 of the SMBENB register.
- 2 Write to the SMBSAR register (bits 9:0) to set the slave address. This is the address to which the SMB responds.
- 3 Write to the SMBCON register to specify which type of addressing is supported (7- or 10-bit by setting bit 3). Enable the SMB in slave-only mode by writing a ‘0’ into bit 6 (SMBSLAVE\_DISABLE) and a ‘0’ to bit 0 (MASTER\_MODE).

**NOTE:** Slaves and masters do not have to be programmed with the same type of addressing 7- or 10-bit address. For instance, a slave can be programmed with 7-bit addressing and a master with 10-bit addressing, and vice versa.

- 4 Enable the SMB by writing a ‘1’ in bit 0 of the SMBENB register.

**NOTE:** Depending on the reset values chosen, steps 2 and 3 may not be necessary because the reset values can be configured. For instance, if the device is only going to be a master, there would be no need to set the slave address because you can configure SMB to have the slave disabled after reset and to enable the master after reset. The values stored are static and do not need to be reprogrammed if the SMB is disabled.

### 29.3.3.2 Slave-Transmitter Operation for a Single Byte

When another SMB master device on the bus addresses the SMB and requests data, the SMB acts as a slave-transmitter and the following steps occur:

- 1 The other SMB master device initiates an SMB transfer with an address that matches the slave address in the SMBSAR register of the SMB.
- 2 The SMB acknowledges the sent address and recognizes the direction of the transfer to indicate that it is acting as a slave-transmitter.
- 3 The SMB asserts the RDREQ interrupt (bit 5 of the SMBINTST register) and holds the SCL line low. It is in a wait state until software responds.

If the RDREQ interrupt has been masked, due to SMBINTM[5] register (MRDREQ bit field) being set to 0, then it is recommended that a hardware and/or software timing routine be used to instruct the CPU to perform periodic reads of the SMBINTST register.

- a Reads that indicate SMBINTST[5] (RDREQ bit field) being set to 1 must be treated as the equivalent of the RDREQ interrupt being asserted.
- b Software must then act to satisfy the SMB transfer.
- c The timing interval used should be in the order of 10 times the fastest SCL clock period the SMB can handle. For example, for 400 kb/s, the timing interval is 25us.

**NOTE:** The value of 10 is recommended here because this is approximately the amount of time required for a single byte of data transferred on the SMB bus.

- 4 If there is any data remaining in the TX FIFO before receiving the read request, then the SMB asserts a TX\_ABRT interrupt (bit 6 of the SMBINTST register) to flush the old data from the TX FIFO.

**NOTE:** Because the SMB's TX FIFO is forced into a flushed/reset state whenever a TX\_ABRT event occurs, it is necessary for software to release the SMB from this state by reading the SMBCTXABT register before attempting to write into the TX FIFO. See register SMBINTST for more details. If the TX\_ABRT interrupt has been masked, due to of SMBINTM[6] register (MTX\_ABRT bit field) being set to 0, then it is recommended that re-using the timing routine (described in the previous step), or a similar one, be used to read the SMBINTST register.

- a Reads that indicate bit 6 (TXABT) being set to 1 must be treated as the equivalent of the TX\_ABRT interrupt being asserted.
  - b There is no further action required from software.
  - c The timing interval used should be similar to that described in the previous step for the SMBINTST[5] register.
- 5 Software writes to the SMBDC register with the data to be written (by writing a '0' in bit 8).
  - 6 Software must clear the RDREQ and TX\_ABRT interrupts (bits 5 and 6, respectively) of the SMBINTST register before proceeding.

If the RDREQ and/or TX\_ABRT interrupts have been masked, then clearing of the



SMBINTST register will have already been performed when either the RDREQ or TXABT bit has been read as 1.

- 7 The SMB releases the SCL and transmits the byte.
- 8 The master may hold the SMB bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

### 29.3.3.3 Slave-Receiver Operation for a Single Byte

When another SMB master device on the bus addresses the SMB and is sending data, the SMB acts as a slave-receiver and the following steps occur:

- 1 The other SMB master device initiates an SMB transfer with an address that matches the SMB's slave address in the SMBSAR register.
- 2 The SMB acknowledges the sent address and recognizes the direction of the transfer to indicate that the SMB is acting as a slave-receiver.
- 3 SMB receives the transmitted byte and places it in the receive buffer.

**NOTE:** If the RX FIFO is completely filled with data when a byte is pushed, then an overflow occurs and the SMB continues with subsequent SMB transfers. Because a NACK is not generated, software must recognize the overflow when indicated by the SMB (by the RXOF bit in the SMBINTST register) and take appropriate actions to recover from lost data. Hence, there is a real time constraint on software to service the RX FIFO before the latter overflow as there is no way to re-apply pressure to the remote transmitting master. You must select a deep enough RX FIFO depth to satisfy the interrupt service interval of their system.

- 4 SMB asserts the RX\_FULL interrupt (SMBINTST[2] register).

If the RX\_FULL interrupt has been masked, due to setting SMBINTM[2] register to 0 or setting SMBTX\_TL to a value larger than 0, then it is recommended that a timing routine (described in "Slave-Transmitter Operation for a Single Byte" on page 57) be implemented for periodic reads of the "SMBST" on page 136 register. Reads of the SMBST register, with bit 3 (RFNE) set at 1, must then be treated by software as the equivalent of the RX\_FULL interrupt being asserted.

- 5 Software may read the byte from the SMBDC register (bits 7:0).
- 6 The other master device may hold the SMB bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

### 29.3.3.4 Slave-Transfer Operation For Bulk Transfers

In the standard SMB protocol, all transactions are single byte transactions and the programmer responds to a remote master read request by writing one byte into the slave's TX FIFO.

When a slave (slave-transmitter) is issued with a read request (RDREQ) from the remote master (master-receiver), at a minimum there should be at least one entry placed into the slave-transmitter's TX FIFO.

SMB is designed to handle more data in the TX FIFO so that subsequent read requests can take that data without raising an interrupt to get more data. Ultimately, this eliminates the possibility of significant latencies being incurred between raising the interrupt for data each time had there been a restriction of having only one entry placed in the TX FIFO.

This mode only occurs when SMB is acting as a slave-transmitter. If the remote master acknowledges the data sent by the slave-transmitter and there is no data in the slave's TX FIFO, the SMB holds the SMB SCL line low while it raises the read request interrupt (RDREQ) and waits for data to be written into the TX FIFO before it can be sent to the remote master.

If the RDREQ interrupt is masked, due to bit 5 (MRDREQ) of the SMBINTST register being set to 0, then it is recommended that a timing routine be used to activate periodic reads of the SMBINTST register. Reads of SMBINTST that return bit 5 (RDREQ) set to 1 must be treated as the equivalent of the RDREQ interrupt referred to in this section.

The RDREQ interrupt is raised upon a read request, and like interrupts, must be cleared when exiting the interrupt service handling routine (ISR). The ISR allows you to either write 1 byte or more than 1 byte into the TX FIFO. During the transmission of these bytes to the master, if the master acknowledges the last byte. Then the slave must raise the RDREQ again because the master is requesting for more data.

### 29.3.4 Disabling SMB

The register SMBENBST is added to allow software to unambiguously determine when the hardware has completely shutdown in response to the SMBENB register being set from 1 to 0.

#### 29.3.4.1 Procedure

- 1 Define a timer interval (tSMB\_poll) equal to the 10 times the signaling period for the highest SMB transfer speed used in the system and supported by SMB. For example, if the highest SMB transfer mode is 400 kb/s, then this tSMB\_poll is 25us.
- 2 Define a maximum time-out parameter, MAX\_T\_POLL\_COUNT, such that if any repeated polling operation exceeds this maximum value, an error is reported.
- 3 Execute a blocking thread/process/function that prevents any further SMB master transactions to be started by software, but allows any pending transfers to be completed.

**NOTE:** This step can be ignored if SMB is programmed to operate as an SMB slave only.

- 4 The variable POLL\_COUNT is initialized to zero.
- 5 Set SMBENB to 0.
- 6 Read the SMBENBST register and test the SMB\_EN bit (bit 0). Increment POLL\_COUNT by one. If POLL\_COUNT >= MAX\_T\_POLL\_COUNT, exit with the relevant error code.
- 7 If SMBENBST[0] is 1, then sleep for tSMB\_poll and proceed to the previous step. Otherwise, exit with a relevant success code.

## 30 Synchronous Serial Interface

### 30.1 Overview

The SSI is a full-duplex synchronous serial interface and can connect to a variety of external analog-to-digital (A/D) converters, audio and telecom codecs, and other devices that use serial protocols for transferring data. The SSI supports National’s Microwire, Texas Instruments Synchronous Serial Protocol (SSP), and Motorola’s Serial Peripheral Interface (SPI) protocol.

The SSI operates in master mode (the attached peripheral functions as a slave) and supports serial bit rates from 7.2 KHz to 54 MHz. Serial data formats may range from 2 to 32 bits in length. The SSI provides 128 entries deep x 32 bits wide transmit and receive data FIFOs.

The FIFOs may be loaded or emptied by the Central Processor Unit (CPU) using programmed I/O, or DMA transfers while receiving or transmitting.

Features:

- 3 protocols support: National's Microwire, TI's SSP, and Motorola's SPI
- Full-duplex or transmit-only or receive-only operation
- Programmable transfer order: MSB first or LSB first
- 128 entries deep x 32 bits wide transmit and receive data FIFOs
- Configurable normal transfer mode or Interval transfer mode
- Programmable clock phase and polarity for Motorola’s SSI format
- Two slave select signal (SSI\_CE\_ / SSI\_CE2\_) supporting up to 2 slave devices
- Back-to-back character transmission/reception mode
- Loop back mode for testing

### 30.2 Pin Description

**Table 30-1 SSI Controller Pins Description**

| Name     | I/O    | Description                                     |
|----------|--------|-------------------------------------------------|
| SSI_CLK  | Output | Serial bit-rate clock                           |
| SSI_CE_  | Output | First slave select enable                       |
| SSI_CE2_ | Output | Second slave select enable                      |
| SSI_GPC  | Output | General purpose control signal to external chip |
| SSI_DT   | Output | Transmit data (serial data out)                 |
| SSI_DR   | Input  | Receive data (serial data in)                   |

SSI\_CLK is the bit-rate clock driven from the SSI to the peripheral. SSI\_CLK is toggled only when data

is actively being transmitted and received.

SSI\_CE\_ or SSI\_CE2\_ are the frame signal, indicating the begin and the end of a serialized data word.

SSI\_DT and SSI\_DR are the Transmit and Receive serial data lines.

SSI\_GPC is general-purpose control signal, synchronized with SSI\_CLK, can be used for LCD control.

### 30.3 Register Description

#### 30.3.1 Register Mapping

The SSI has the following registers: one data, two control, one status, one bit-rate control, and two interval control registers. The table lists these registers.

**Table 30-2 SSI Serial Port Registers**

| Name     | RW | Reset Value | Address Offset | Access Size |
|----------|----|-------------|----------------|-------------|
| SSIDR0   | RW | 0x??        | 0x10043000     | 32          |
| SSICR00  | RW | 0x0000      | 0x10043004     | 32          |
| SSICR10  | RW | 0x00087860  | 0x10043008     | 32          |
| SSISR0   | RW | 0x00000098  | 0x1004300C     | 32          |
| SSIITR0  | RW | 0x0000      | 0x10043010     | 16          |
| SSIICR0  | RW | 0x00        | 0x10043014     | 8           |
| SSIGR0   | RW | 0x0000      | 0x10043018     | 16          |
| SSIRCNT0 | RW | 0x00000000  | 0x1004301C     | 32          |
|          |    |             |                |             |
| SSIDR1   | RW | 0x??        | 0x10044000     | 32          |
| SSICR01  | RW | 0x0000      | 0x10044004     | 32          |
| SSICR11  | RW | 0x00087860  | 0x10044008     | 32          |
| SSISR1   | RW | 0x00000098  | 0x1004400C     | 32          |
| SSIITR1  | RW | 0x0000      | 0x10044010     | 16          |
| SSIICR1  | RW | 0x00        | 0x10044014     | 8           |
| SSIGR1   | RW | 0x0000      | 0x10044018     | 16          |
| SSIRCNT1 | RW | 0x00000000  | 0x1004401C     | 32          |

**NOTE:** There two SSI controller. SSI0 whose base address is 0x100430xx and SSI1 whose base address is 0x100440xx.

#### 30.3.2 SSI Data Register (SSIDR)

| SSIDR |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         | 0x10043000, 0x10044000 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit   | 31      | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16      | 15                     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|       | D31-D17 |    |    |    |    |    |    |    |    |    |    |    |    |    |    | GPC/D16 | D15-D0                 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST   | ?       | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?       | ?                      | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits  | Name    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW |
|-------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:17 | D31-D17 | D31-D17 to be written to/read from Transmit/Receive FIFO.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | R  |
| 16    | GPC/D16 | This bit can be used as normal data bus bit 16 or GPC bit alternatively. When it is used as normal data bus bit, it's readable / writable; when SSI_GPC is used, it is GPC bit for SSI_GPC pin output and it's write-only.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW |
| 15:0  | D15-D0  | <p>D15-D0 to be written to/read from Transmit/Receive FIFO.</p> <p>When the transfer frame length is less than 17-bit, received data is automatically right justified in the receive-FIFO and the upper unused bits are filled with '0'. For transmission, the upper unused bits of the data written into SSIDR is ignored by the transmit logic. (<b>NOTE:</b> "upper unused bits" does not include the SSIDR.GPC bit.</p> <p>National microwire format includes format 1 and format2, when national microwire format 2 is selected, Bit 16 of SSIDR is defined as read/write operation judge bit, if it is 0, bit 15~0 represent one read command; if it is 1, bit 15~0 represent one write command and following is the written data. So the maximum length of one command (is defined in MCOM) is 16, the maximum length of one written or read data (is defined in FLEN) can be 17.</p> <p>Transmit-FIFO only contain one read operation command once, or one write operation command and its data once, after transmit-FIFO is empty, next command can be filled in transmit-FIFO.</p> | RW |

### 30.3.3 SSI Control Register0 (SSICR0)

| SSICR0 |          |    |    |    |    |    |    |    |    |    |    |    |         |         |      |     | 0x10043004, 0x10044004 |      |      |      |       |       |        |      |          |       |        |        |        |        |   |   |
|--------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|---------|------|-----|------------------------|------|------|------|-------|-------|--------|------|----------|-------|--------|--------|--------|--------|---|---|
| Bit    | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19      | 18      | 17   | 16  | 15                     | 14   | 13   | 12   | 11    | 10    | 9      | 8    | 7        | 6     | 5      | 4      | 3      | 2      | 1 | 0 |
|        | Reserved |    |    |    |    |    |    |    |    |    |    |    | TENDIAN | RENDIAN | SSIE | TIE | RIE                    | TEIE | REIE | LOOP | RFINE | RFINC | EACLUN | FSEL | Reserved | VRCNT | TFMODE | TFLUSH | RFLUSH | DISREV |   |   |
| RST    | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0       | 0    | 0   | 0                      | 0    | 0    | 0    | 0     | 0     | 0      | 0    | 0        | 0     | 0      | 0      | 0      | 0      | 0 | 0 |

| Bits  | Name     | Description                          | RW |
|-------|----------|--------------------------------------|----|
| 31:20 | Reserved | Writing has no effect, read as zero. | R  |

| 19:18 | TENDIAN        | Transfer endian<br>0: Most significant byte in a word first, most significant bit in a byte first.<br>1: Most significant byte in a word first, least significant bit in a byte first.<br>2: Least significant byte in a word first, least significant bit in a byte first.<br>3: Least significant byte in a word first, most significant bit in a byte first. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
|-------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|--------------------------|---|---|-------------------------------------------------------------------------------------|---|---|------------------|---|---|----------------|
| 17:16 | RENDIAN        | Receive endian<br>0: Most significant byte in a word first, most significant bit in a byte first.<br>1: Most significant byte in a word first, least significant bit in a byte first.<br>2: Least significant byte in a word first, least significant bit in a byte first.<br>3: Least significant byte in a word first, most significant bit in a byte first   | RW                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 15    | SSIE           | This bit is used to enable/disable SSI module. 0: disable; 1: enable. Clearing SSIE will not reset SSI FIFO, SSICR0, SSICR1, SSIGR, SSIITR and SSIICR automatically. Software should ensure the FIFOs/registers are properly configured and be flush/reset manually when necessary before enabling SSI.                                                         | RW                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 14    | TIE            | This bit enables/disables the transmit-FIFO half-empty interrupt TXI.<br>0: disable; 1: enable.                                                                                                                                                                                                                                                                 | RW                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 13    | RIE            | This bit enables/disables the receive-FIFO half-full interrupt RXI.<br>0: disable; 1: enable.                                                                                                                                                                                                                                                                   | RW                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 12    | TEIE           | This bit enables/disables the transmit-error interrupt TEI.<br>0: disable; 1: enable.                                                                                                                                                                                                                                                                           | RW                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 11    | REIE           | This bit enables/disables the receive-error interrupt REI.<br>0: disable; 1: enable.                                                                                                                                                                                                                                                                            | RW                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 10    | LOOP           | Used for test purpose. In loop mode, the output of SSI transmit shift register is connected to input of SSI receive shift register internally. The data received should be the same as the data transmitted. And do not output any valid signals on the pins.<br>0: normal SSI mode; 1: LOOP mode.                                                              | RW                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 9     | RFINE          | Receive finish control enable.<br>0: disable; 1: enable.<br>For SSICR1.FMAT = B'10 (National Microwire format 1 is selected), SSICR0.RFINE must be 0.                                                                                                                                                                                                           | The receive finish condition list below: <table border="1" data-bbox="762 1444 1305 1825"> <thead> <tr> <th>RFINE</th> <th>RFINC</th> <th>Receive Finish Condition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>x</td> <td>Same as transmit completion condition (transmit-fifo is empty and SSICR1.UNFIN = 0)</td> </tr> <tr> <td>1</td> <td>0</td> <td>Receive continue</td> </tr> <tr> <td>1</td> <td>1</td> <td>Receive finish</td> </tr> </tbody> </table> | RFINE | RFINC | Receive Finish Condition | 0 | x | Same as transmit completion condition (transmit-fifo is empty and SSICR1.UNFIN = 0) | 1 | 0 | Receive continue | 1 | 1 | Receive finish |
| RFINE | RFINC          | Receive Finish Condition                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 0     | x              | Same as transmit completion condition (transmit-fifo is empty and SSICR1.UNFIN = 0)                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 1     | 0              | Receive continue                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 1     | 1              | Receive finish                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 8     | RFINC*         | Receive finish control bit.<br>0: receive continue<br>1: receive finished                                                                                                                                                                                                                                                                                       | RW                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |
| 7     | EACL<br>R<br>N | 0: don't auto clear under flag, software clear under<br>1: software auto clear under flag when tffifo don't empty                                                                                                                                                                                                                                               | RW                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |                          |   |   |                                                                                     |   |   |                  |   |   |                |

|   |          |                                                                                                                                                                           |    |
|---|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 6 | FSEL     | This bit sets the frame signal to be used for slave select. The unselected frame signal always output invalid level.<br>0: SSI_CE_ is selected<br>1: SSI_CE2_ is selected | RW |
| 5 | Reserved | Writing has no effect, read as zero.                                                                                                                                      | R  |
| 4 | VRCNT    | Receive counter (RCNT) valid flag.<br>0: SSIRCNT.RCNT will be ignored.<br>1: SSIRCNT.RCNT will be used.                                                                   | RW |
| 3 | TFMODE   | 0: new fifo empty mode<br>1: old fifo empty mode                                                                                                                          | RW |
| 2 | TFLUSH   | Flush the transmit FIFO when set to 1.<br>Always return 0 when read.                                                                                                      | RW |
| 1 | RFLUSH   | Flush the receive FIFO when set to 1.<br>Always return 0 when read.                                                                                                       | RW |
| 0 | DISREV   | Receive function enable.<br>0: enable.<br>1: disable.                                                                                                                     | RW |

**NOTE:**

- \*: When transmitting finished or for receive-only operation, transmit function can be disabled and this bit is used to control receiving completion, and the SSI will consume less power.

When the finish condition is set, the receiving will complete after present character is completely shifted in, then the SSI will stop the SSI\_CLK and negate the SSI\_CE\_ / SSI\_CE2\_ if necessary. To make sure present transfer is completed, user must read and get SSISR.END = 1 (or SSISR.BUSY = 0).

**30.3.4 SSI Control Register1 (SSICR1)**

| SSICR1 |       |       |       |          |       |       |          |      |      |      |      |      |          |     |     | 0x10043008, 0x10044008 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|--------|-------|-------|-------|----------|-------|-------|----------|------|------|------|------|------|----------|-----|-----|------------------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| Bit    | 31    | 30    | 29    | 28       | 27    | 26    | 25       | 24   | 23   | 22   | 21   | 20   | 19       | 18  | 17  | 16                     | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|        | FRMHL | TFVCK | TCKFI | Reserved | ITFRM | UNFIN | Reserved | FMAT | TTRG | MCOM | RTRG | FLEN | Reserved | PHA | POL |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST    | 0     | 0     | 0     | 0        | 0     | 0     | 0        | 0    | 0    | 0    | 0    | 0    | 1        | 0   | 0   | 0                      | 0  | 1  | 1  | 1  | 1  | 0  | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bits  | Name              | Description                                                                             | RW            |
|-------|-------------------|-----------------------------------------------------------------------------------------|---------------|
| 31:30 | FRMHL             | Frame valid level select, FRMHL [1: 0] correspond to SSI_CE2_ and SSI_CE_ respectively. | RW            |
|       | <b>FRMHL[1:0]</b> | <b>Description</b>                                                                      |               |
|       | 00                | SSI_CE_ is low level valid and                                                          | Initial value |

|            |                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | SSI_CE2_ is low level valid                                  |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
|------------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|--|------------|-------------|--|----|--------------------------------------|---------------|----|-------------------------------------|--|----|-------------------------------------|--|----|-------------------------------------|--|----|
|            |                                      | 01                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | SSI_CE_ is high level valid and SSI_CE2_ is low level valid  |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
|            |                                      | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | SSI_CE_ is low level valid and SSI_CE2_ is high level valid  |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
|            |                                      | 11                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | SSI_CE_ is high level valid and SSI_CE2_ is high level valid |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 29:28      | TFVCK                                | <p>Time from frame valid to clock start. When TFVCK = B'00, the time is fixed half or one SSI_CLK cycle according to SSICR1.POL and SSICR1.PHA configuration.</p> <p>For SSICR1.FMAT = B'01, SSICR1.TFVCK is ignored.</p> <table border="1"> <thead> <tr> <th>TFVCK[1:0]</th> <th>Description</th> <th></th> </tr> </thead> <tbody> <tr> <td>00</td> <td>half or one SSI_CLK cycle delay time</td> <td>Initial value</td> </tr> <tr> <td>01</td> <td>1 more SSI_CLK cycle delay is added</td> <td></td> </tr> <tr> <td>10</td> <td>2 more SSI_CLK cycle delay is added</td> <td></td> </tr> <tr> <td>11</td> <td>3 more SSI_CLK cycle delay is added</td> <td></td> </tr> </tbody> </table>     |                                                              |  | TFVCK[1:0] | Description |  | 00 | half or one SSI_CLK cycle delay time | Initial value | 01 | 1 more SSI_CLK cycle delay is added |  | 10 | 2 more SSI_CLK cycle delay is added |  | 11 | 3 more SSI_CLK cycle delay is added |  | RW |
| TFVCK[1:0] | Description                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 00         | half or one SSI_CLK cycle delay time | Initial value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 01         | 1 more SSI_CLK cycle delay is added  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 10         | 2 more SSI_CLK cycle delay is added  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 11         | 3 more SSI_CLK cycle delay is added  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 27:26      | TCKFI                                | <p>Time from clock stop to frame invalid. When TCKFI = B'00, the time is fixed one SSI_CLK or half SSI_CLK cycle according to SSICR1.POL and SSICR1.PHA configuration.</p> <p>For SSICR1.FMAT = B'01, SSICR1.TFVCK is ignored.</p> <table border="1"> <thead> <tr> <th>TCKFI[1:0]</th> <th>Description</th> <th></th> </tr> </thead> <tbody> <tr> <td>00</td> <td>half or one SSI_CLK cycle delay</td> <td>Initial value</td> </tr> <tr> <td>01</td> <td>1 more SSI_CLK cycle delay is added</td> <td></td> </tr> <tr> <td>10</td> <td>2 more SSI_CLK cycle delay is added</td> <td></td> </tr> <tr> <td>11</td> <td>3 more SSI_CLK cycle delay is added</td> <td></td> </tr> </tbody> </table> |                                                              |  | TCKFI[1:0] | Description |  | 00 | half or one SSI_CLK cycle delay      | Initial value | 01 | 1 more SSI_CLK cycle delay is added |  | 10 | 2 more SSI_CLK cycle delay is added |  | 11 | 3 more SSI_CLK cycle delay is added |  | RW |
| TCKFI[1:0] | Description                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 00         | half or one SSI_CLK cycle delay      | Initial value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 01         | 1 more SSI_CLK cycle delay is added  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 10         | 2 more SSI_CLK cycle delay is added  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 11         | 3 more SSI_CLK cycle delay is added  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                              |  |            |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 25         | Reserved                             | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                              |  | R          |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 24         | ITFRM                                | <p>Frame during interval, selects if the Frame (SSI_CE_ /SSI_CE2_) signal is negated or not during interval time at Interval Mode (SSICR1.FMAT = B'00 and SSIITR.IVLTM ≠ H'0000). It's ignored at Normal Mode.</p> <p>0: SSI_CE_ /SSI_CE2_ de-asserts during interval time at Interval Mode<br/> 1: SSI_CE_ /SSI_CE2_ keeps asserted during interval time at Interval Mode</p>                                                                                                                                                                                                                                                                                                                  |                                                              |  | RW         |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |
| 23         | UNFIN                                | <p>This bit controls whether the SSI finishes transmission or wait for data filling (underrun happen) after all data in transmit-FIFO are sent out during transfer. This bit must be cleared to 0 when SSICR1.FMAT = B'01. (TI's SSP format)</p> <p>0: TxFIFO empty means end of transmission<br/> 1: Transmission didn't finish when transmit-FIFO is empty, SSI underrun error would occur and SSI waits for data filling; SSI_CLK and SSI_CE_ /SSI_CE2_ keeps asserted, SSI_CLK stop at the current level</p> <p><b>NOTE:</b> For TxFIFO empty before any transfer after SSI enabled, if</p>                                                                                                 |                                                              |  | RW         |             |  |    |                                      |               |    |                                     |  |    |                                     |  |    |                                     |  |    |



|       |                             | SSICR1.UNFIN = 1 or SSICR0.RFINE = 0, SSI will wait till TxFIFO isn't empty then start to transfer and no underrun error will occur; if SSICR1.UNFIN = 0 and SSICR0.RFINE = 1, after transmit-FIFO become empty, SSI will start a receive-only transfer.                                                                                                                                                                                                                                                                          |      |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
|-------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------|---------------|----|-----------------------|---------------|----|-----------------|--|----|-----------------------------|--|----|-----------------------------|--|----|
| 22    | Reserved                    | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | R    |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 21:20 | FMAT                        | <p>These bits set the operating transfer format.</p> <table border="1"> <thead> <tr> <th>FMAT</th> <th>Description</th> <th>Initial value</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>Motorola's SPI format</td> <td>Initial value</td> </tr> <tr> <td>01</td> <td>TI's SSP format</td> <td></td> </tr> <tr> <td>10</td> <td>National Microwire 1 format</td> <td></td> </tr> <tr> <td>11</td> <td>National Microwire 2 format</td> <td></td> </tr> </tbody> </table>                                                        | FMAT | Description | Initial value | 00 | Motorola's SPI format | Initial value | 01 | TI's SSP format |  | 10 | National Microwire 1 format |  | 11 | National Microwire 2 format |  | RW |
| FMAT  | Description                 | Initial value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 00    | Motorola's SPI format       | Initial value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 01    | TI's SSP format             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 10    | National Microwire 1 format |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 11    | National Microwire 2 format |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 19:16 | TTRG                        | <p>These bits define the transmit-FIFO half-empty threshold value, which when equal or less characters left in transmit-FIFO, the SSISR.TFHE will be set to '1'.</p> <p>0000: less than or equal to 1<br/>n: less than or equal to nx8</p>                                                                                                                                                                                                                                                                                        | RW   |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 15:12 | MCOM                        | <p>Defines the length of command.</p> <p>Only used when SSICR1.FMAT = b'10 or b'11 (National Microwire format 1 or 2 is selected).</p> <p>The command length is (MCOM + 1).</p> <p>Initial value is b'0111.</p>                                                                                                                                                                                                                                                                                                                   | RW   |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 11:8  | RTRG                        | <p>These bits define the receive-FIFO half-full threshold value, which when equal or more characters received in receive-FIFO, the SSISR.RFHF will be set to '1'.</p> <p>0000: more than or equal to 1<br/>n: more than or equal to nx8</p>                                                                                                                                                                                                                                                                                       | RW   |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 7:3   | FLEN                        | <p>These bits set the bit length of every character to be transmitted/received. The maximum data length can be configured is 32 bits. For data length longer than 17 bits (multiples of the SSICR1.FLEN configured length), the software should ensure properly processing.</p> <p><b>When SSI_GPC pin is used, the FLEN shouldn't be configured as B'01111 (17-bit data).</b></p> <p>When TI SSP mode is selected (FMAT = 2'b01), 2-bit data length (FLEN = 0) isn't supported.</p> <p>The bit length of one data = FLEN + 2</p> | RW   |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 2     | Reserved                    | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | R    |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |
| 1     | PHA                         | <p>This bit sets the phase of the SSI_CLK from the beginning of a data frame for Motorola's SPI format (SSICR1.FMAT = B'00).</p> <p>0: The leading edge of SSI_CLK is used to sample data from SSI_DR after the SSI_CE_ /SSI_CE2_ goes valid, it is initial value<br/>1: The leading edge of SSI_CLK is used to drive data onto SSI_DT after the SSI_CE_ /SSI_CE2_ goes valid</p>                                                                                                                                                 | RW   |             |               |    |                       |               |    |                 |  |    |                             |  |    |                             |  |    |

|   |     |                                                                                                                                                                         |    |
|---|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 0 | POL | SSI_CLK polarity for Motorola's SPI format during idle state.<br>(SSICR1.FMAT = B'00).<br>0: SSI_CLK keeps low level when idle<br>1: SSI_CLK keeps high level when idle | RW |
|---|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|

### 30.3.5 SSI Status Register (SSISR)

| SSISR |          |    |    |    |    |    |    |    |           |    |    |    |    |    |    | 0x1004300C, 0x1004400C |           |    |    |    |    |    |   |   |     |      |     |     |      |      |      |      |   |
|-------|----------|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|------------------------|-----------|----|----|----|----|----|---|---|-----|------|-----|-----|------|------|------|------|---|
| Bit   | 31       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23        | 22 | 21 | 20 | 19 | 18 | 17 | 16                     | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6    | 5   | 4   | 3    | 2    | 1    | 0    |   |
|       | Reserved |    |    |    |    |    |    |    | TFIFO-NUM |    |    |    |    |    |    |                        | RFIFO-NUM |    |    |    |    |    |   |   | END | BUSY | TFF | RFE | TFHE | RFHF | UNDR | OVER |   |
| RST   | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0                      | 0         | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0   | 1    | 0   | 0   | 1    | 1    | 0    | 0    | 0 |

| Bits  | Name      | Description                                                                                                                                                                                                                                                                                                     | RW |
|-------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:24 | Reserved  | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                            | R  |
| 23:16 | TFIFO-NUM | These bits indicate the Characters Number in Transmit-FIFO.                                                                                                                                                                                                                                                     | R  |
| 15:8  | RFIFO-NUM | These bits indicate the Characters Number in Transmit-FIFO.                                                                                                                                                                                                                                                     | R  |
| 7     | END       | This bit indicates transfer end status. It is the inverse of SSISR.BUSY when transfer is in process, but it'll keep cleared at interval time before transfer is completed. It'll be set when transfer finished.                                                                                                 | R  |
| 6     | BUSY      | This bit indicates SSI's working status.<br>0: SSI is idle or at interval time<br>1: Transmission and/or reception is in process                                                                                                                                                                                | R  |
| 5     | TFF       | This bit denotes transmit-FIFO is full or not.<br>0: Transmit-FIFO is not full<br>1: Transmit-FIFO is full                                                                                                                                                                                                      | R  |
| 4     | RFE       | This bit denotes receive-FIFO is empty or not.<br>0: Receive-FIFO is not empty<br>1: Receive-FIFO is empty                                                                                                                                                                                                      | R  |
| 3     | TFHE      | This bit denotes whether the characters number in transmit-FIFO being less or equal to SSICR1.TTRG.<br>0: The data in transmit-FIFO is more than the condition set by SSICR1.TTRG<br>1: The data in transmit-FIFO meets the condition set by SSICR1.TTRG, If SSICR0.TIE = 1, it will generate SSI TXI interrupt | R  |
| 2     | RFHF      | This bit denotes whether the characters number in receive-FIFO being more or equal to the number set by SSICR1.RTRG.<br>0: The data in receive-FIFO is less than the condition set by SSICR1.RTRG<br>1: The data in receive-FIFO meets the condition set by                                                     | R  |

|   |      |                                                                                                                                                                                                                                                                                                                                                          |    |
|---|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|   |      | SSICR1.RTRG, If SSICR0.RIE = 1, it will generate SSI RXI interrupt                                                                                                                                                                                                                                                                                       |    |
| 1 | UNDR | Transmit-FIFO underrun status. When underrun happens, SSI set this bit and keeps the current status of SSI_CLK and SSI_CE_/SSI_CE2_, waiting for transmit-FIFO filling.<br>0: Underrun has not occurred<br>1: Underrun has occurred, when SSICR0.TEIE is set, it will generate SSI TEI interrupt. Write '0' to clear this bit, writing '1' has no effect | RW |
| 0 | OVER | Receive-FIFO overrun status, new received data will lose.<br>0: Overrun has not occurred<br>1: Overrun has occurred; When SSICR0.REIE is set, it will generate SSI REI interrupt. Write '0' to clear this bit, writing '1' has no effect                                                                                                                 | RW |

### 30.3.6 SSI Interval Time Control Register (SSIITR)

|        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                        |       |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|--------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------------------|-------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| SSIITR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x10043010, 0x10044010 |       |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| Bit    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15                     | 14    | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CNTCLK                 | IVLTM |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                        | 0     | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW |
|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 15   | CNTCLK | Counting clock source select.<br>0: Use SSI bit clock (SSI_CLK) as the interval counter clock source<br>1: Use 32K clock as the interval counter clock source                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW |
| 14:0 | IVLTM  | Interval time set, set the cycle number of counting clock source for desired interval time. When SSIITR.IVLTM = 0x0000, normal mode is selected, and SSIITR.CNTCLK and SSICR are ignored. When SSIITR.IVLTM ≠ 0x0000, interval mode is selected. The interval time is calculated as follows:<br>$Interval\ time \approx [CNTCLK\ clock\ period] * [Value\ of\ IVLTM]$<br><br>The actual interval time is as follow:<br>When SSIITR.CNTCLK = 0:<br>$Interval\ time = [CNTCLK\ clock\ period] * [Value\ of\ IVLTM] + 3 * device\_clock\ period$<br><br>When SSIITR.CNTCLK = 1:<br>$Interval\ time \geq [CNTCLK\ clock\ period] * [Value\ of\ IVLTM + 1] + 1 * device\_clock\ period;$ | RW |



### 30.3.9 SSI Receive Counter Register (SSIRCNT)

|                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                               |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------------------|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>SSIRCNT</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x1004301C, 0x1004401C</b> |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>     | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16                            | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                               | RCNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                             | 0    | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                                                              | RW |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:16 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                     | R  |
| 15:0  | RCNT     | Number of data to be received.<br>If SSICR1.UNFIN = 0 and SSICR0.RFINE = 1, after transmit-FIFO become empty, SSI will start a receive-only transfer. If SSICR0.VRCNT = 1, a special internal counter will be increased automatically. When the value of the internal counter equals to the RCNT, the receiving operation will be stopped automatically. | RW |

### 30.4 Functional Description

Serial data is transferred between the processor and external peripheral through FIFO buffers in the SSI. Data transfers to system memory are handled by either the CPU (using programmed I/O) or by DMA. Operation is full duplex - separate buffers and serial data paths permit simultaneous transfers to and from the external peripheral.

Programmed I/O transmits and receives data directly between the CPU and the transmit/receive FIFO's. The DMA controller transfers data during transmit and receive operations between memory and the FIFO's.

Transmit data is written by the CPU or DMA to the SSI's transmit FIFO. The SSI then takes the data from the FIFO, serializes it, and transmits it via the SSI\_DT signal to the peripheral. Data from the peripheral is received via the SSI\_DR signal, converted to parallel words and is stored in the Receive FIFO. Read operations automatically target the receive FIFO, while write operations write data to the transmit FIFO. Both the transmit and receive FIFO buffers are 128 entries deep by 17 bits wide. As the received data fills the receive FIFO, a programmable threshold triggers an interrupt to the Interrupt Controller. If enabled, an interrupt service routine responds by identifying the source of the interrupt and then performs one or several read operations from the inbound (receive) FIFO buffer.

### 30.5 Data Formats

Four signals are used to transfer data between the processor and external peripheral. The SSI supports three formats: Motorola SPI, Texas Instruments SSP, and National Microwire. Although they

have the same basic structure the three formats have significant differences, as described below:

- 1 SSI\_CE\_/SSI\_CE2\_ varies for each protocol as follows:
  - For SPI and Microwire formats, SSI\_CE\_/SSI\_CE2\_ functions as a chip select to enable the external device (target of the transfer), and is held active-low during the data transfer.
  - For SSP format, this signal is pulsed high for one serial bit-clock period at the start of each frame.
  
- 2 SSI\_CLK varies for each protocol as follows:
  - For Microwire, both transmit and receive data sources switch data on the falling edge of SSI\_CLK, and sample incoming data on the rising edge.
  - For SSP, transmit and receive data sources switch data on the rising edge of SSI\_CLK, and sample incoming data on the falling edge.
  - For SPI, the user has the choice of which edge of SSI\_CLK to use for switching outgoing data, and for sampling incoming data. In addition, the user can move the phase of SSI\_CLK, shifting its active state one-half period earlier or later at the start and end of a frame.

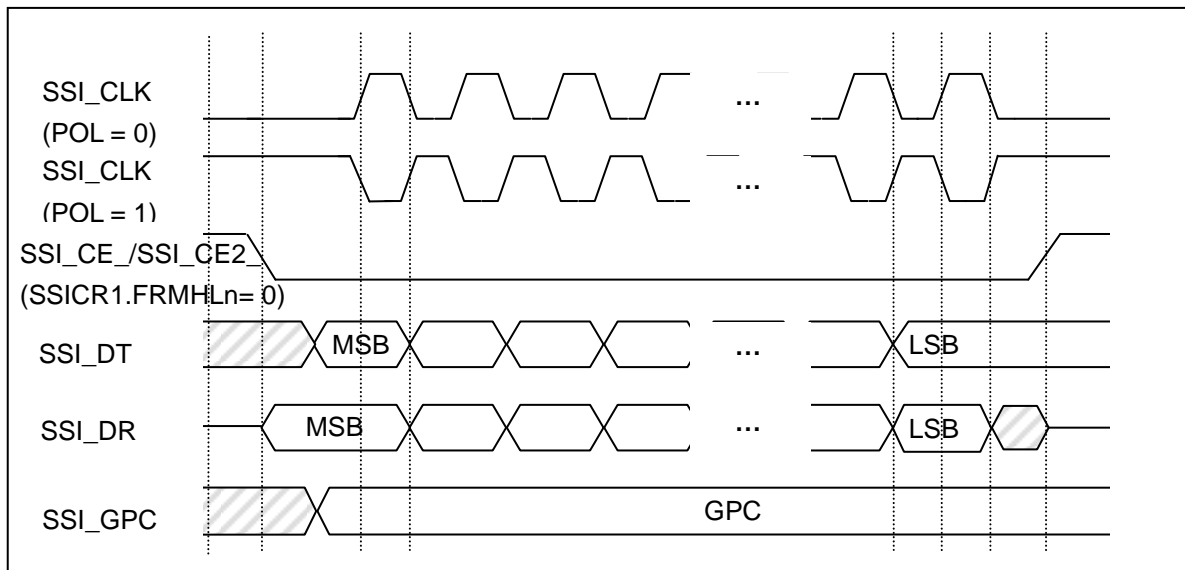
While SSP and SPI are full-duplex protocols, Microwire uses a half-duplex master-slave messaging protocol. At the start of a frame, a 1 or 2-byte control message is transmitted from the controller to the peripheral. The peripheral does not send any data. The peripheral interprets the message and, if it is a READ request, responds with requested data, one clock after the last bit of the requesting message.

The serial clock (SSI\_CLK) only toggles during an active frame. At other times it is held in an inactive or idle state, as defined by its specified protocol.

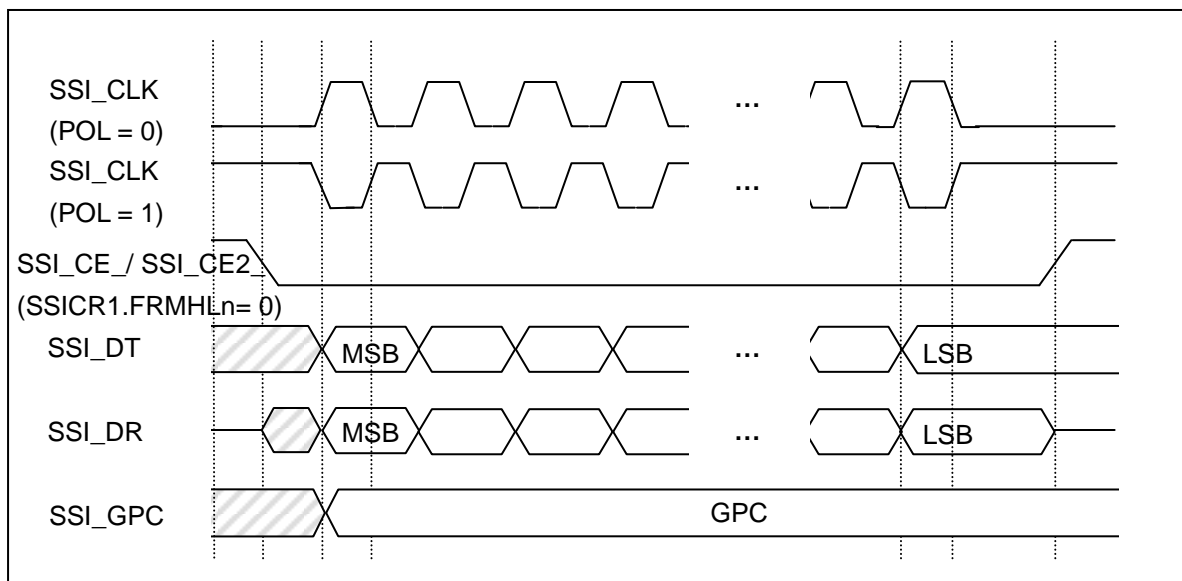
### 30.5.1 Motorola's SPI Format Details

#### 30.5.1.1 General Single Transfer Formats

The figures below show the timing of general single transfer format.



**Figure 30-1 SPI Single Character Transfer Format (PHA = 0)**



**Figure 30-2 SPI Single Character Transfer Format (PHA = 1)**

For SSICR1.PHA = 0, when SSICR1.TFVCK = B'00, hardware ensures the first clock edge appears one SSI\_CLK period after SSI\_CE\_ / SSI\_CE2\_ goes valid; when SSICR1.TCKFI = B'00, hardware ensures the SSI\_CE\_ / SSI\_CE2\_ negated half SSI\_CLK period after last clock change edge; when SSICR1.TFVCK ≠ B'00 or SSICR1.TCKFI ≠ B'00, 1/2/3 more clock cycles are inserted.

For SSICR1.PHA = 1, when SSICR1.TFVCK = B'00, hardware ensures the first clock edge appears half SSI\_CLK period after SSI\_CE\_ / SSI\_CE2\_ goes valid; when SSICR1.TCKFI = B'00, hardware ensures the SSI\_CE\_ / SSI\_CE2\_ negated one SSI\_CLK period after last clock change edge; when SSICR1.TFVCK ≠ B'00 or SSICR1.TCKFI ≠ B'00, 1/2/3 more clock cycles are inserted.

Data is sampled from SSI\_DR at every rising edge (when PHA = 0, POL = 0 or PHA = 1, POL = 1) or at every falling edge (when PHA = 0, POL = 1 or PHA = 1, POL = 0). According to SPI protocol, input data on SSI\_DR should be stable at every sample clock edge.

Drive data onto SSI\_DT at every rising edge (when PHA = 0, POL = 1 or PHA = 1, POL = 0) or at every falling edge (when PHA = 0, POL = 0 or PHA = 1, POL = 1).

### 30.5.1.2 Back-to-Back Transfer Formats

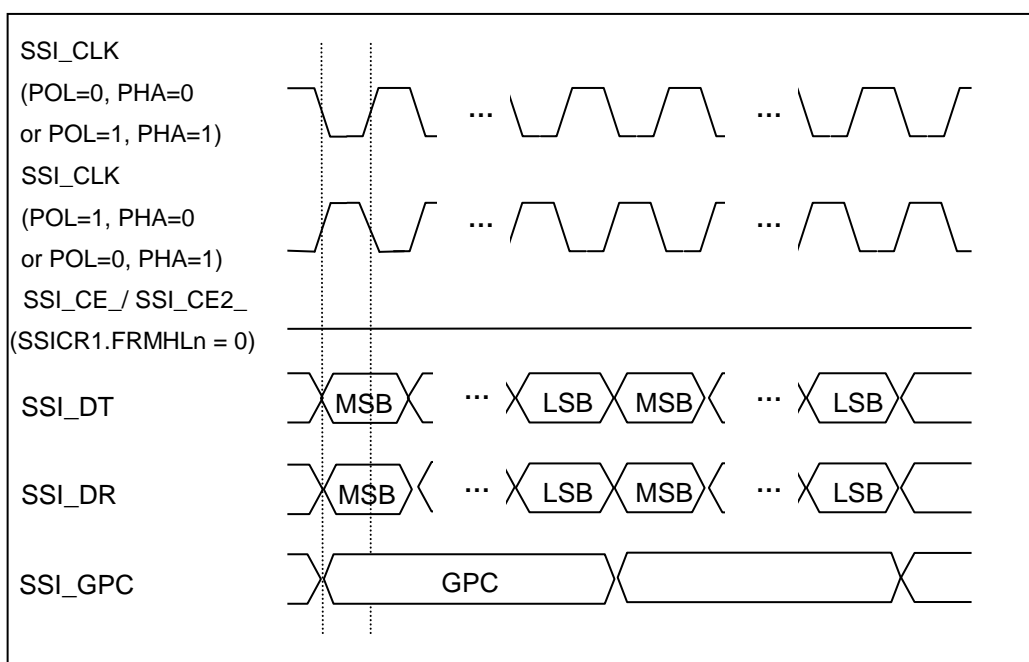


Figure 30-3 SPI Back-to-Back Transfer Format

For Motorola's SPI format transfers those continuous characters are exchanged during SSI\_CE\_ / SSI\_CE2\_ being valid, the timing is illustrated in the figure (SSICR1.LFST = 0).

Back-to-back transfer is performed as transmit-only/full-duplex operation when transmit-FIFO is not empty before the completion of the last character's transfer or performed as receive-only operation.

### 30.5.1.3 Frame Interval Mode Transfer Format

When in interval mode (SSIITR.IVLTM ≠ '0'), SSI always wait for an interval time (SSIITR.IVLTM), transfer fixed number of characters (SSIICR), then repeats the operation.

When SSICR0.RFINE = 1, if transmit-FIFO is still empty after the interval time, receive-only transfer will occur.



During interval-wait time, SSI stops SSI\_CLK, and when SSICR1.ITFRM = 0 it negates the SSI\_CE\_ / SSI\_CE2\_, when SSICR1.ITFRM = 1 it keeps asserting the SSI\_CE\_ / SSI\_CE2\_.

For transfers finished with transmit-FIFO empty, if the SSI transmit-FIFO is empty before fixed number of characters being loaded to transfer (SSICR1.UNFIN must be 1), then the SSI will set SSISR.UNDR = 1; if enabled, it'll send out a SSI underrun interrupt. At the same time, SSI will hold the SSI\_CE\_ / SSI\_CE2\_ and SSI\_CLK signals at current status and wait for the transmit-FIFO filling. The SSI will continue transfer after transmit-FIFO being filled. The SSI always stops after completion of fixed number of characters' transfer (SSICR1.UNFIN must be 0) with transmit-FIFO empty.

For transfers finished by SSICR0.RFINC being valid set, the SSI will stop after finished current character transfer and needn't wait for a whole completion of fixed number of characters' transfer.

Two Interval transfer mode are illustrated in the following figures. In these timing diagram, SSICR1.PHA = 0, SSICR1.POL = 0 and SSIICR = 0.

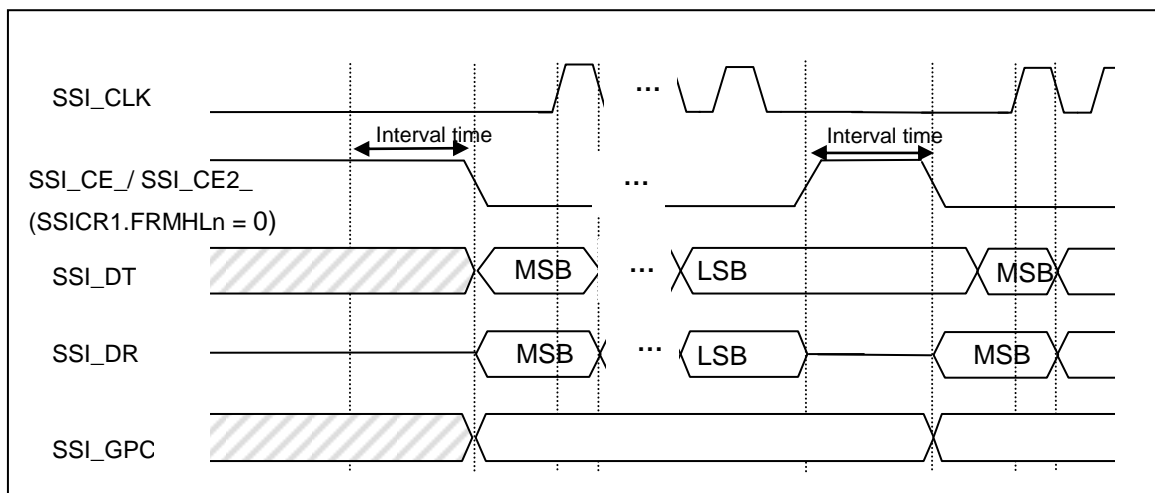


Figure 30-4 SPI Frame Interval Mode Transfer Format (ITFRM = 0, LFST = 0)

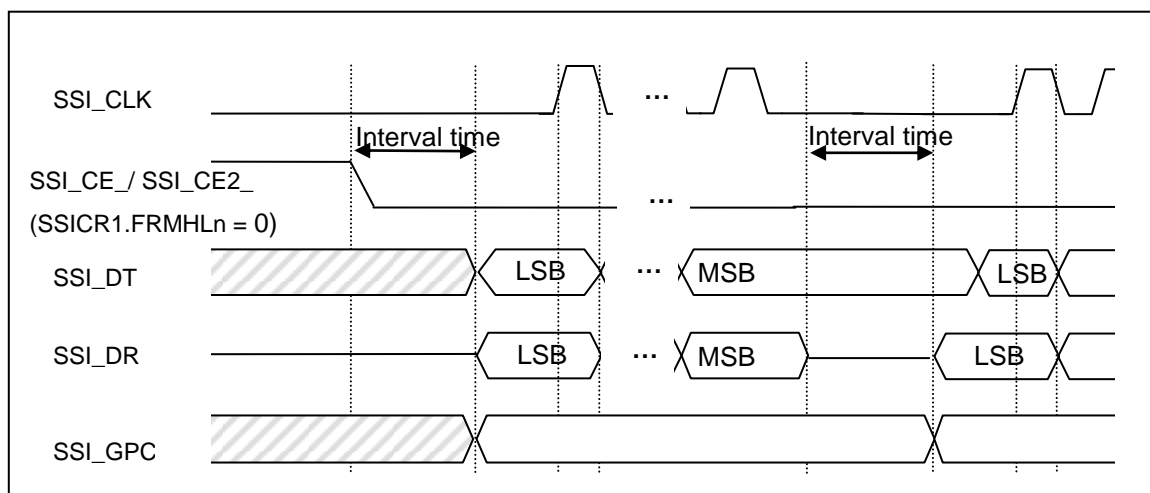


Figure 30-5 SPI Frame Interval Mode Transfer Format (ITFRM = 1, LFST = 1)

### 30.5.2 TI's SSP Format Details

In this format, each transfer begins with SSI\_CE\_ pulsed high for one SSI\_CLK period. Then both master and slave drive data at SSI\_CLK's rising edge and sample data at the falling edge. Data are transferred with MSB first or LSB first. At the end of the transfer, SSI\_DT retains the value of the last bit sent through the next idle period.

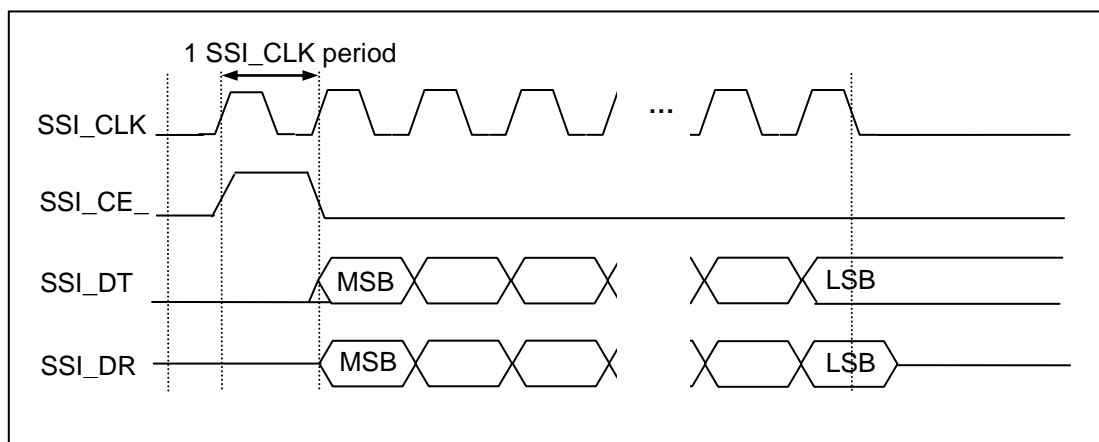


Figure 30-6 TI's SSP Single Transfer Format

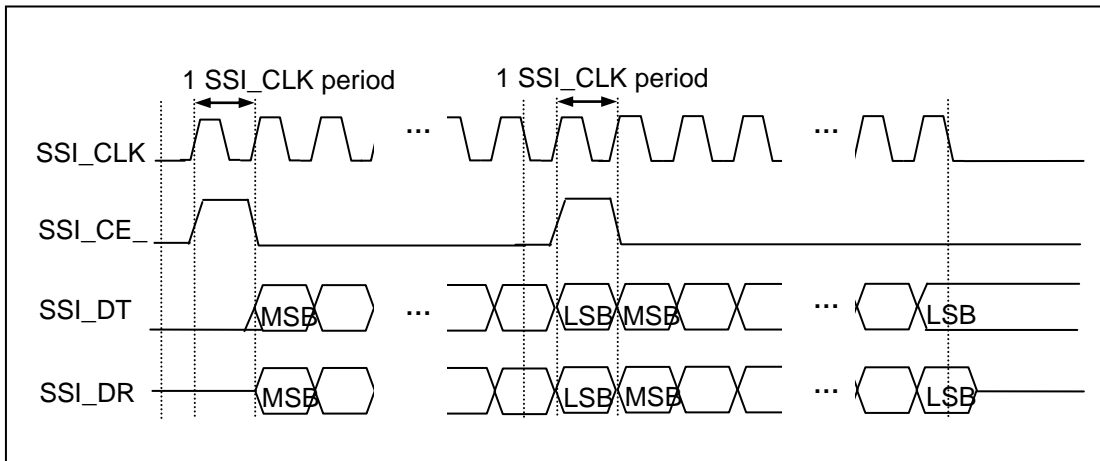


Figure 30-7 TI's SSP Back-to-back Transfer Format

### 30.5.3 National Microwire Format Details

It supports format 1 and format 2.

|             | Format | Drive Edge   | Sample Edge  |
|-------------|--------|--------------|--------------|
| NM Format 1 | Master | Falling Edge | Rising Edge  |
|             | Slave  | Falling Edge | Rising Edge  |
| NM Format 2 | Master | Falling Edge | Falling Edge |
|             | Slave  | Rising Edge  | Rising Edge  |

If format 1 is selected, both master and slave drive data at SSI\_CLK falling edge and sample data at the rising edge.

If format 2 is selected, master drive and sample data at SSI\_CLK falling edge, slave drive and sample data at SSI\_CLK rising edge.

SSI\_CLK goes high midway through the command's MSB (or LSB) and continues to toggle at the bit rate. One bit clock (format 1) or half bit clock (format 2) period after the last command bit, the external slave must return the serial data requested, with MSB first (or LSB first) on SSI\_DR.

SSI\_CE\_ / SSI\_CE2 de-asserts high half clock (SSI\_CLK) period (and 1/2/3 additional clock periods) later.

Format 1 support back-to-back transfer, the start and end of back-to-back transfers are similar to those of a single transfer. However, SSI\_CE\_ / SSI\_CE2 remains asserted throughout the transfer. The end of a character data on SSI\_DR is immediately followed by the start of the next command byte on SSI\_DT.

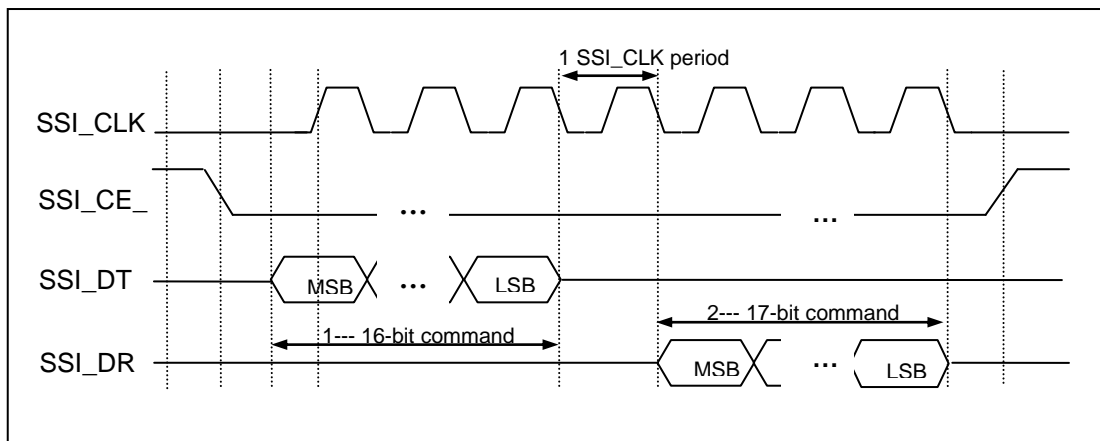


Figure 30-8 National Microwire Format 1 Single Transfer

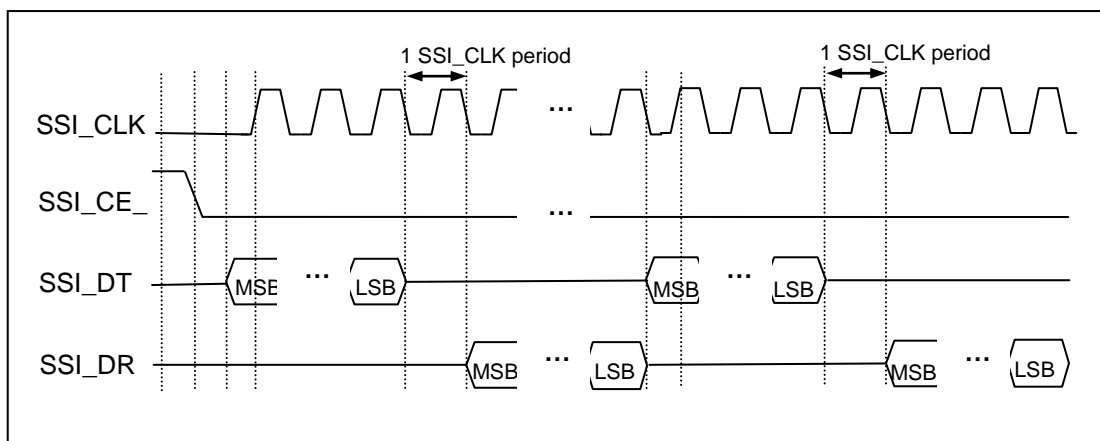


Figure 30-9 National Microwire Format 1 Back-to-back Transfer

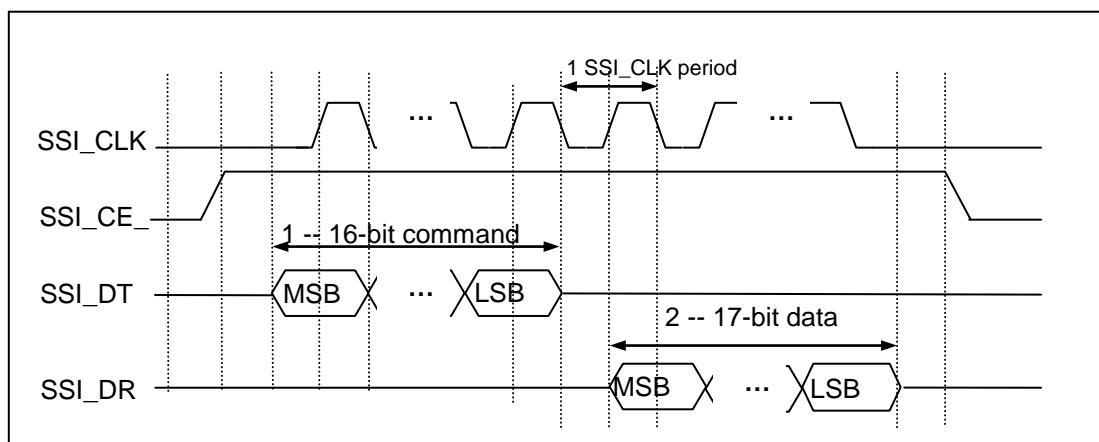


Figure 30-10 National Microwire Format 2 Read Timing

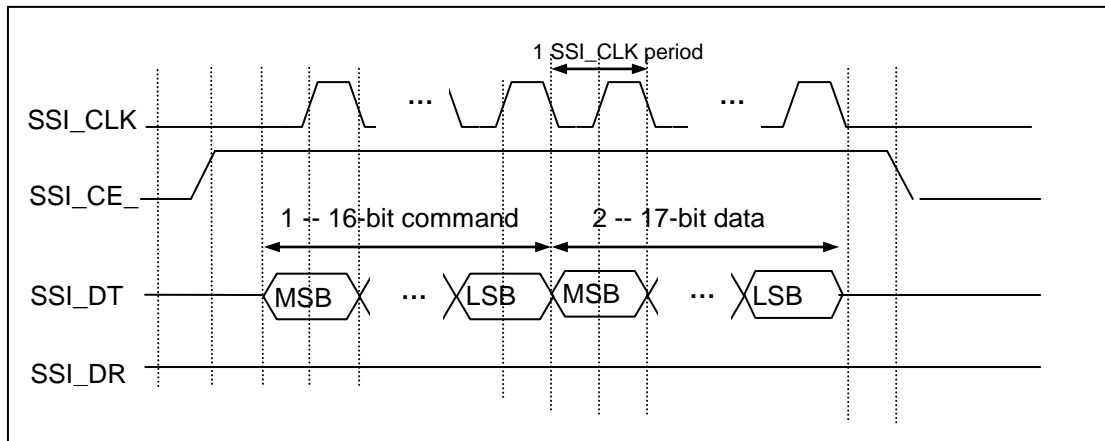


Figure 30-11 National Microwire Format 2 Write Timing

### 30.6 Interrupt Operation

In SSI, there are TXI, RXI, TEI and REI total 4 interrupts, all these interrupts are combined together to make one SSI interrupt, which can be masked by writing '1' into corresponding mask bit in INTC interrupt mask register (IMR).

Table 30-3 SSI Interrupts

| Operation | Condition                    | Flag Bit   | Mask Bit    | Interrupt  | DMAC Activation |
|-----------|------------------------------|------------|-------------|------------|-----------------|
| Transmit  | T-FIFO is half-empty or less | SSISR.TFHE | SSICR0.TIE  | <b>TXI</b> | Possible        |
|           | Transmit underrun error      | SSISR.UNDR | SSICR0.TEIE | <b>TEI</b> | Impossible      |
| Receive   | R-FIFO is half-full or more  | SSISR.RFHF | SSICR0.RIE  | <b>RXI</b> | Possible        |
|           | Receive overrun error        | SSISR.OVER | SSICR0.REIE | <b>REI</b> | Impossible      |

Either SSISR.TFHE or SSISR.RFHF can activate DMA transferring when corresponding individual interrupt mask bit in SSICR0 is cleared (masked) and DMA is enabled and configured.

## 31 UART Interface

### 31.1 Overview

This chapter describes the universal asynchronous receiver/transmitter (UART) serial ports. There are four UARTs: All UARTs use the same programming model. Each of the serial ports can operate in interrupt based mode or DMA-based mode.

The Universal asynchronous receiver/transmitter (UART) is compatible with the 16550-industry standard and can be used as slow infrared asynchronous interface that conforms to the Infrared Data Association (IrDA) serial infrared specification 1.1.

### Features

- Full-duplex operation
- 5-, 6-, 7- or 8-bit characters with optional no parity or even or odd parity and with 1, 1½, or 2 stop bits
- 64x8 bit transmit FIFO and 64x11bit receive FIFO
- Independently controlled transmit, receive (data ready or timeout), line status interrupts
- Internal diagnostic capability Loopback control and break, parity, overrun and framing-error is provided
- Separate DMA requests for transmit and receive data services in FIFO mode
- Supports modem flow control by software or hardware
- Slow infrared asynchronous interface that conforms to IrDA specification

#### 31.1.1 Pin Description

**Table 31-1 UART Pins Description**

| Name | Type   | Description                                 |
|------|--------|---------------------------------------------|
| RxD  | Input  | Receive data input                          |
| TxD  | Output | Transmit data output                        |
| CTS_ | Input  | Clear to Send — Modem Transmission enabled  |
| RTS_ | Output | Request to Send — UART Transmission request |

**NOTE:** UART3, UART2, UART1, UART0 support RxD, TxD, RTS\_, CTS\_.

### 31.2 Register Descriptions

All UART register 32-bit access address is physical address. When ULCR.DLAB is 0, URBR, UTHR and UIER can be accessed; When ULCR.DLAB is 1, UDLLR and UDLHR can be accessed.

### 31.2.1 Register Mapping

There are 5 UART controllers in chip, name UART0, UART1, UART2, UART3, UART4 respectively.

The base address of UART0 is 0x10030000.

The base address of UART1 is 0x10031000.

The base address of UART2 is 0x10032000.

The base address of UART3 is 0x10033000.

The base address of UART4 is 0x10034000.

**Table 31-2 UART Registers Description**

| Name  | Description                            | RW | Reset Value | Offset Address | Access Size |
|-------|----------------------------------------|----|-------------|----------------|-------------|
| URBR  | UART Receive Buffer Register           | R  | 0x??        | 0x000          | 8           |
| UTHR  | UART Transmit Hold Register            | W  | 0x??        | 0x000          | 8           |
| UDLLR | UART Divisor Latch Low Register        | RW | 0x00        | 0x000          | 8           |
| UDLHR | UART Divisor Latch High Register       | RW | 0x00        | 0x004          | 8           |
| UIER  | UART Interrupt Enable Register         | RW | 0x00        | 0x004          | 8           |
| UIIR  | UART Interrupt Identification Register | R  | 0x01        | 0x008          | 8           |
| UFCR  | UART FIFO Control Register             | W  | 0x00        | 0x008          | 8           |
| ULCR  | UART Line Control Register             | RW | 0x00        | 0x00c          | 8           |
| UMCR  | UART Modem Control Register            | RW | 0x00        | 0x010          | 8           |
| ULSR  | UART Line Status Register              | R  | 0x00        | 0x014          | 8           |
| UMSR  | UART Modem Status Register             | R  | 0x00        | 0x018          | 8           |
| USPR  | UART Scratchpad Register               | RW | 0x00        | 0x01c          | 8           |
| ISR   | Infrared Selection Register            | RW | 0x00        | 0x020          | 8           |
| UMR   | UART M Register                        | RW | 0x00        | 0x024          | 8           |
| UACR  | UART Add Cycle Register                | RW | 0x00        | 0x028          | 16          |
| URCR  | UART RXFIFO Counter Register           | R  | 0x00        | 0x040          | 8           |
| UTCR  | UART TXFIFO Counter Register           | R  | 0x00        | 0x044          | 8           |

### 31.2.2 UART Receive Buffer Register (URBR)

The read-only URBR is corresponded to one level 11bit buffer in non-FIFO mode and a 32x11bit FIFO that holds the character(s) received by the UART. Bits in URBR are right justified when being configured to use fewer than eight bits, and the rest of most significant data bits are zeroed and the most significant three bits of each buffer are the status for the character in the buffer. If ULSR.DRY is 0, don't read URBR, otherwise wrong operation may occur.

|            |             |  |  |  |  |  |  |  |  |  |  |                         |  |  |  |  |  |  |  |  |  |   |   |   |   |   |   |   |   |
|------------|-------------|--|--|--|--|--|--|--|--|--|--|-------------------------|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|---|---|
|            | <b>URBR</b> |  |  |  |  |  |  |  |  |  |  | <b>0x000 (DLAB = 0)</b> |  |  |  |  |  |  |  |  |  |   |   |   |   |   |   |   |   |
| <b>Bit</b> |             |  |  |  |  |  |  |  |  |  |  |                         |  |  |  |  |  |  |  |  |  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

|      |      |                               |  |  |  |  |  |  |  |  |  |  |  |  |  |  | URBR |   |   |   |   |   |   |   |   |    |
|------|------|-------------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------|---|---|---|---|---|---|---|---|----|
| RST  |      |                               |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ?    | ? | ? | ? | ? | ? | ? | ? | ? | ?  |
| Bits | Name | Description                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |   |   |   |   |   |   |   |   | RW |
| 7:0  | URBR | 8-bit UART receive read data. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |   |   |   |   |   |   |   |   | R  |

**31.2.3 UART Transmit Hold Register (UTHR)**

The write-only UTHR is corresponded to one level 8 bit buffer in non-FIFO mode and a 32x8bit FIFO in FIFO mode that holds the data byte(s) to be transmitted next.

|     |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                  |   |   |   |   |   |   |   |
|-----|------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------------|---|---|---|---|---|---|---|
|     | UTHR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x000 (DLAB = 0) |   |   |   |   |   |   |   |
| Bit |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7                | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|     |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | UTHR             |   |   |   |   |   |   |   |
| RST |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ?                | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description                          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | RW |
|------|------|--------------------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----|
| 7:0  | UTHR | 8-bit UART transmit write hold data. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | W  |

**31.2.4 UART Divisor Latch Low/High Register (UDLLR / UDLHR)**

UART Divisor Latch registers, UDLLR/UDLHR together compose the divisor for the programmable baud rate generator that can take the UART device clock and divide it by 1 to (2<sup>16</sup> - 1).

The UART device source clock is EXCLK or EXCLK/2 that is determined by CPCCR.ECS. UDLHR/UDLLR stores the high/low 8-bit of the divisor respectively. Load these divisor latches during initialization to ensure that the baud rate generator operates properly. If both Divisor Latch registers are 0, the 16X clock stops.

If you don't set UMR and UACR, UART will work at normal mode with the specified frequency. The relationship between baud rate and the value of Divisor is shown by the formula when UMR and UACR are not set:

$$\text{Baud Rate} = (\text{UART device clock}) / (16 * \text{Divisor})$$

|     |       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                         |   |   |   |   |   |   |   |
|-----|-------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------------------|---|---|---|---|---|---|---|
|     | UDLLR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x000 (DLAB = 1)        |   |   |   |   |   |   |   |
| Bit |       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7                       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|     |       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Divisor Latch Low 8-bit |   |   |   |   |   |   |   |
| RST |       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                       | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



|              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                         |                          |   |   |   |   |   |   |  |
|--------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------------------|--------------------------|---|---|---|---|---|---|--|
| <b>UDLHR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x004 (DLAB = 1)</b> |                          |   |   |   |   |   |   |  |
| <b>Bit</b>   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7                       | 6                        | 5 | 4 | 3 | 2 | 1 | 0 |  |
|              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                         | Divisor Latch High 8-bit |   |   |   |   |   |   |  |
| <b>RST</b>   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                       | 0                        | 0 | 0 | 0 | 0 | 0 | 0 |  |

### 31.2.5 UART Interrupt Enable Register (UIER)

The UART Interrupt Enable Register (UIER) contains the interrupt enable bits for the five types of interrupts (receive data ready, timeout, line status, and transmit data request, and modem status) that set a value in UIIR.

|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                         |          |   |   |       |      |       |       |       |
|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------------------|----------|---|---|-------|------|-------|-------|-------|
| <b>UIER</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x004 (DLAB = 0)</b> |          |   |   |       |      |       |       |       |
| <b>Bit</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7                       | 6        | 5 | 4 | 3     | 2    | 1     | 0     |       |
|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                         | Reserved |   |   | RTOIE | MSIE | RLSIE | TDRIE | RDRIE |
| <b>RST</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0                       | 0        | 0 | 0 | 0     | 0    | 0     | 0     |       |

| Bits | Name     | Description                                                                                                                                                                                                                                                               | RW |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 7:5  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                      | R  |
| 4    | RTOIE    | <b>Receive Timeout Interrupt Enable.</b><br>0: Disable the receive timeout interrupt<br>1: Enable the receive timeout interrupt<br>Timeout means the URDR (FIFO mode) is not empty but no character has received for a period of time T: T (bits) = 4 X Word length + 12. | RW |
| 3    | MSIE     | <b>Modem Status Interrupt Enable.</b><br>0: Disable the modem status interrupt<br>1: Enable the modem status interrupt                                                                                                                                                    | RW |
| 2    | RLSIE    | <b>Receive Line Status Interrupt Enable.</b><br>0: Disable receive line status interrupt<br>1: Enable receive line status interrupt                                                                                                                                       | RW |
| 1    | TDRIE    | <b>Transmit Data Request Interrupt Enable.</b><br>0: Disable the transmit data request interrupt<br>1: Enable the transmit data request interrupt                                                                                                                         | RW |
| 0    | RDRIE    | <b>Receive Data Ready Interrupt Enable.</b><br>0: Disable the receive data ready interrupt<br>1: Enable the receive data ready interrupt                                                                                                                                  | RW |

### 31.2.6 UART Interrupt Identification Register (UIIR)

The read-only UART Interrupt Identification Register (UIIR) records the prioritized pending interrupt source information. Its initial value after power-on reset is 0x01.

|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |               |              |                 |   |             |   |   |               |   |
|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---------------|--------------|-----------------|---|-------------|---|---|---------------|---|
| <b>UIIR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |               | <b>0x008</b> |                 |   |             |   |   |               |   |
| <b>Bit</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |               | 7            | 6               | 5 | 4           | 3 | 2 | 1             | 0 |
|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>FFMSEL</b> |              | <b>Reserved</b> |   | <b>INID</b> |   |   | <b>INPEND</b> |   |
| <b>RST</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |               | 0            | 0               | 0 | 0           | 0 | 0 | 0             | 1 |

| Bits  | Name                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW   |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
|-------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------|-------|--------------|-------|-----------------------|-------|--------------------|-------|---------------------|-------|----------|-------|----------|-------|------------------|-------|----------|---|
| 7:6   | FFMSEL                | <b>FIFO Mode Select.</b><br>0b00: Non-FIFO mode<br>0b01: Reserved<br>0b10: Reserved<br>0b11: FIFO mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | R    |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 5:4   | Reserved              | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | R    |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 3:1   | INID                  | <b>Interrupt Identifier.</b><br>These bits identify the current highest priority pending interrupt.<br><table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>INID</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b000</td> <td>Modem Status</td> </tr> <tr> <td>0b001</td> <td>Transmit Data Request</td> </tr> <tr> <td>0b010</td> <td>Receive Data Ready</td> </tr> <tr> <td>0b011</td> <td>Receive Line Status</td> </tr> <tr> <td>0b100</td> <td>Reserved</td> </tr> <tr> <td>0b101</td> <td>Reserved</td> </tr> <tr> <td>0b110</td> <td>Receive Time Out</td> </tr> <tr> <td>0b111</td> <td>Reserved</td> </tr> </tbody> </table><br>See Table 31-3 for details. | INID | Description | 0b000 | Modem Status | 0b001 | Transmit Data Request | 0b010 | Receive Data Ready | 0b011 | Receive Line Status | 0b100 | Reserved | 0b101 | Reserved | 0b110 | Receive Time Out | 0b111 | Reserved | R |
| INID  | Description           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 0b000 | Modem Status          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 0b001 | Transmit Data Request |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 0b010 | Receive Data Ready    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 0b011 | Receive Line Status   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 0b100 | Reserved              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 0b101 | Reserved              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 0b110 | Receive Time Out      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 0b111 | Reserved              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |
| 0     | INPEND                | <b>Interrupt Pending.</b><br>0: interrupt is pending<br>1: No interrupt pending                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | R    |             |       |              |       |                       |       |                    |       |                     |       |          |       |          |       |                  |       |          |   |

**Table 31-3 UART Interrupt Identification Register Description**

| UIIR.INID | Interrupt Set/Clear Cause |                     |                                                                               |                                                            |
|-----------|---------------------------|---------------------|-------------------------------------------------------------------------------|------------------------------------------------------------|
|           | Priority                  | Type                | Source                                                                        | Clear Condition                                            |
| 0b0001    | —                         | None                | No pending interrupt                                                          | —                                                          |
| 0b0110    | 1st Highest               | Receive Line Status | Overrun, Parity, Frame Error, Break Interrupt, and FIFO Error (DMA mode only) | Reading ULSR or empty all the error characters in DMA mode |

|        |             |                       |                                                                                                |                                                                                    |
|--------|-------------|-----------------------|------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| 0b0100 | 2nd Highest | Receive Data Ready    | FIFO mode: Trigger threshold was reached<br>Non-FIFO mode: URBR full                           | FIFO mode: Reading URBR till below trigger threshold.<br>Non-FIFO mode: Empty URBR |
| 0b1100 | 2nd Highest | Receive Timeout       | FIFO mode only: URBR not empty but no data read in for a period of time                        | Reset receive buffer by setting UFCR.RFRT to 1 or Reading URBR                     |
| 0b0010 | 3rd Highest | Transmit Data Request | FIFO mode: Empty location in UTHR equal to half or more than half<br>Non-FIFO mode: UTHR empty | FIFO mode: Data number in UTHR more than half<br>Non-FIFO mode: Writing UTHR       |
| 0b0000 | 4th Highest | Modem Status          | Modem CTS_ pin status change                                                                   | Reading UMSR                                                                       |

### 31.2.7 UART FIFO Control Register (UFCR)

The write-only register UFCR contains the control bits for receive and transmit FIFO.

|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |                 |            |            |             |             |            |   |   |              |
|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|-----------------|------------|------------|-------------|-------------|------------|---|---|--------------|
| <b>UFCR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |                 |            |            |             |             |            |   |   | <b>0x008</b> |
| <b>Bit</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             | 7               | 6          | 5          | 4           | 3           | 2          | 1 | 0 |              |
|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>RDTR</b> | <b>Reserved</b> | <b>UME</b> | <b>DME</b> | <b>TFRT</b> | <b>RFRT</b> | <b>FME</b> |   |   |              |
| <b>RST</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             | 0               | 0          | 0          | 0           | 0           | 0          | 0 | 0 |              |

| Bits | Name     | Description                                                                                                                                                                                     | RW |
|------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 7:6  | RDTR     | <b>Receive Buffer Data Number Trigger.</b><br>These bits are used to select the trigger level for the receive data ready interrupt in FIFO mode.<br>0b00: 1<br>0b01: 16<br>0b10: 32<br>0b11: 60 | W  |
| 5    | Reserved | Writing has no effect, read as zero.                                                                                                                                                            | R  |
| 4    | UME      | <b>UART Module Enable.</b><br>0: Disable UART<br>1: Enable UART                                                                                                                                 | W  |
| 3    | DME      | <b>DMA Mode Enable.</b><br>0: Disable DMA mode<br>1: Enable DMA mode                                                                                                                            | W  |
| 2    | TFRT     | <b>Transmit Holding Register Reset.</b>                                                                                                                                                         | W  |

|   |      |                                                                                                         |   |
|---|------|---------------------------------------------------------------------------------------------------------|---|
|   |      | 0: Not reset<br>1: Reset transmit FIFO                                                                  |   |
| 1 | RFRT | <b>Receive Buffer Reset.</b><br>0: Not reset<br>1: Reset receive FIFO                                   | W |
| 0 | FME  | <b>FIFO Mode Enable.</b><br>Set this bit before the trigger levels.<br>0: non-FIFO mode<br>1: FIFO mode | W |

### 31.2.8 UART Line Control Register (ULCR)

The ULCR defines the format for UART data transmission.

|            |             |  |  |  |  |  |  |  |  |  |  |  |  |  |              |     |       |      |      |      |     |   |   |   |   |
|------------|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------|-----|-------|------|------|------|-----|---|---|---|---|
|            | <b>ULCR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x00C</b> |     |       |      |      |      |     |   |   |   |   |
| <b>Bit</b> |             |  |  |  |  |  |  |  |  |  |  |  |  |  |              |     |       | 7    | 6    | 5    | 4   | 3 | 2 | 1 | 0 |
|            |             |  |  |  |  |  |  |  |  |  |  |  |  |  | DLAB         | SBK | STPAR | PARM | PARE | SBLS | WLS |   |   |   |   |
| <b>RST</b> |             |  |  |  |  |  |  |  |  |  |  |  |  |  |              |     |       | 0    | 0    | 0    | 0   | 0 | 0 | 0 | 0 |

| Bits | Name  | Description                                                                                                                                                                                                                                  | RW |
|------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 7    | DLAB  | <b>Divisor Latch Access Bit.</b><br>0: Enable to access URBR, UTHR or UIER<br>1: Enable to access UDLLR or UDLHR                                                                                                                             | W  |
| 6    | SBK   | <b>Set Break.</b><br>Causes a break condition (at least one 0x00 data) to be transmitted to the receiving UART. Acts only on the TXD pin and has no effect on the transmit logic.<br>0: No effect on TXD output<br>1: Forces TXD output to 0 | W  |
| 5    | STPAR | <b>Sticky Parity.</b><br>Setting this bit forces parity location to be opposite of PARM bit when PARE is 1 (it is ignored when PARE is 0).<br>0: Disable Sticky parity<br>1: Enable Sticky parity (opposite of PARM bit)                     | W  |
| 4    | PARM  | <b>Parity Odd/Even Mode Select.</b><br>If PARE = 0, PARM is ignored.<br>0: Odd parity<br>1: Even parity                                                                                                                                      | W  |
| 3    | PARE  | <b>Parity Enable.</b><br>Enables a parity bit to be generated on transmission or checked on reception.                                                                                                                                       | W  |

|     |      |                                                                                                                                                                                                                                                             |   |
|-----|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|     |      | 0: No parity<br>1: Parity                                                                                                                                                                                                                                   |   |
| 2   | SBLS | <b>Stop Bit Length Select.</b><br>Specifies the number of stop bits transmitted and received in each character. When receiving, the receiver checks only the first stop bit.<br>0: 1 stop bit<br>1: 2 stop bits, except for 5-bit character then 1-1/2 bits | W |
| 1:0 | WLS  | <b>Word Length Select.</b><br>0b00: 5-bit character<br>0b01: 6-bit character<br>0b10: 7-bit character<br>0b11: 8-bit character                                                                                                                              | W |

### 31.2.9 UART Line Status Register (ULSR)

The read-only ULSR indicates status information during the data transfer. Receive error information in ULSR[4:1] remains set until software reads ULSR and it must be read before the error character is read.

| ULSR |  | 0x014 |  |  |  |  |  |       |      |      |    |      |       |      |     |
|------|--|-------|--|--|--|--|--|-------|------|------|----|------|-------|------|-----|
| Bit  |  |       |  |  |  |  |  | 7     | 6    | 5    | 4  | 3    | 2     | 1    | 0   |
|      |  |       |  |  |  |  |  | FIFOE | TEMP | TDRQ | BI | FMER | PARER | OVER | DRY |
| RST  |  |       |  |  |  |  |  | 0     | 1    | 1    | 0  | 0    | 0     | 0    | 0   |

| Bits | Name  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RW |
|------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 7    | FIFOE | <b>FIFO Error Status. (FIFO mode only)</b><br>FIFOE is set when there is at least one kind of receive error (parity, frame, overrun, break) for any of the characters in receive buffer. FIFOE is reset when all error characters are read out of the buffer.<br><br>During DMA transfer, the error interrupt generates when FIFOE is 1, and no receive DMA request generates even when data in receive buffer reaches the trigger threshold until all the error characters are read out. In non-DMA mode, FIFOE set does not generate error interrupt.<br><br>0: No error data in receive buffer or non-FIFO mode<br>1: One or more error character in receive buffer | R  |
| 6    | TEMP  | <b>Transmit Holding Register Empty.</b><br>Set when both UTHR and shift register are empty. It is cleared when either the UTHR or the shift register contains a data character.<br><br>0: There is data in the transmit shifter and UTHR                                                                                                                                                                                                                                                                                                                                                                                                                               | R  |

|   |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |   |
|---|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|   |       | 1: All the data in the transmit shifter and UTHR has been shifted out                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |   |
| 5 | TDRQ  | <p><b>Transmit Data Request.</b><br/>Set when UTHR has half or more empty location (FIFO mode) or empty (non-FIFO mode).</p> <p>When both UIER.TDRIE and TDRQ are 1, transmit data request interrupt generates or during DMA transfer, DMA request to the DMA controller generates when UIER.TDRIE is 0 and TDRQ is 1.</p> <p>0: There is one (non-FIFO mode) or more than half data (FIFO mode) in UTHR<br/>1: None data (non-FIFO mode) or half or less than half data (FIFO mode) in UTHR</p>                                                                             | R |
| 4 | BI    | <p><b>Break Interrupt.</b><br/>BI is set when the received data input is held low for longer than a full-word transmission time (the total time of start bit + data bits + parity bit + stop bits). BI is cleared when the processor reads the ULSR. In FIFO mode, only one character equal to 0x00 is loaded into the FIFO regardless of the length of the break condition. BI shows the break condition for the character at the front of the FIFO, not the most recently received character.</p> <p>0: No break signal has been received<br/>1: Break signal received</p> | R |
| 3 | FMER  | <p><b>Framing Error.</b><br/>Set when the bit following the last data bit or parity bit is detected to be 0. If the ULCR had been set for two or one and half stop bits, the other stop bits are not checked except the first one. In FIFO mode, FMER shows a framing error for the character at the front of the receive buffer, not for the most recently received character.</p> <p>Cleared when the processor reads the ULSR.</p> <p>0: No framing error<br/>1: Invalid stop bit has been detected</p>                                                                   | R |
| 2 | PARER | <p><b>Parity Error.</b><br/>Indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. PARER is set upon detection if a parity error and is cleared when the processor reads the ULSR. In FIFO mode, PARER shows a parity error for the character at the front of the FIFO, not the most recently received character.</p> <p>0: No parity error<br/>1: Parity error has occurred</p>                                                                                                                | R |
| 1 | OVER  | <p><b>Overrun Error.</b><br/>Set when both receive buffer and shifter are full and new data is received which will be lost.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                              | R |

|   |     |                                                                                                                                                                                                                                                                                                                                     |   |
|---|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|   |     | Cleared when the processor reads the ULSR.<br>0: No data has been lost<br>1: Receive data has been lost                                                                                                                                                                                                                             |   |
| 0 | DRY | <b>Data Ready.</b><br>Set when a complete incoming character has been received into the Receive Buffer registers. DRY is cleared when the receive buffer is read (non-FIFO mode) or when the buffer is empty or when the buffer is reset by setting UFCR.RFRT to 1.<br>0: No data has been received<br>1: Data is available in URBR | R |

### 31.2.10 UART Modem Control Register (UMCR)

The UMCR uses the modem control pins RTS\_ and CTS\_ to control the interface with a modem or data set. UMCR also controls the loopback mode. Loopback mode must be enabled before the UART is enabled.

|     |      |  |  |  |  |  |  |  |  |  |  |  |  |  |      |     |          |      |          |     |          |   |   |
|-----|------|--|--|--|--|--|--|--|--|--|--|--|--|--|------|-----|----------|------|----------|-----|----------|---|---|
|     | UMCR |  |  |  |  |  |  |  |  |  |  |  |  |  |      |     |          |      | 0x010    |     |          |   |   |
| Bit |      |  |  |  |  |  |  |  |  |  |  |  |  |  |      | 7   | 6        | 5    | 4        | 3   | 2        | 1 | 0 |
|     |      |  |  |  |  |  |  |  |  |  |  |  |  |  | MDCE | FCM | Reserved | LOOP | Reserved | RTS | Reserved |   |   |
| RST |      |  |  |  |  |  |  |  |  |  |  |  |  |  |      | 0   | 0        | 0    | 0        | 0   | 0        | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RW |
|------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 7    | MDCE     | <b>Modem Control Enable.</b><br>0: Modem function is disabled<br>1: Modem function is enabled                                                                                                                                                                                                                                                                                                                                                                                                                                    | W  |
| 6    | FCM      | <b>Flow Control Mode.</b><br>0: Flow control by software<br>1: Flow control by hardware                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |
| 5    | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | R  |
| 4    | LOOP     | <b>Loop Back.</b><br>This bit is used for diagnostic testing of the UART. When LOOP is 1, TXD output pin is set to a logic 1 state, RXD is disconnected from the pin, and the output of the transmitter shifter register is looped back into the receiver shift register input internally, similar to CTS_ and RTS_ pins and the RTS bit of the UMCR is connected to CTS bit of UMSR respectively. Loopback mode must be selected before the UART is enabled.<br><br>0: Normal operation mode<br>1: Loopback-mode UART operation | W  |

|     |          |                                                                                                                         |   |
|-----|----------|-------------------------------------------------------------------------------------------------------------------------|---|
| 3:2 | Reserved | Writing has no effect, read as zero.                                                                                    | R |
| 1   | RTS      | <b>Request To Send.</b><br>This bit can control the RTS_ output state.<br>0: RTS_ force to high<br>1: RTS_ force to low | W |
| 0   | Reserved | Writing has no effect, read as zero.                                                                                    | R |

### 31.2.11 UART Modem Status Register (UMSR)

The read-only UMSR provides the current state of the control lines from the modem to the processor. They are cleared when the processor reads UMSR.

|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          |              |     |          |   |      |   |   |   |
|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----------|--------------|-----|----------|---|------|---|---|---|
| <b>UMSR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          | <b>0x018</b> |     |          |   |      |   |   |   |
| <b>Bit</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          | 7            | 6   | 5        | 4 | 3    | 2 | 1 | 0 |
|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved |              | CTS | Reserved |   | CCTS |   |   |   |
| <b>RST</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          | 0            | 0   | 0        | 0 | 0    | 0 | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                    | RW |
|------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 7:5  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                           | R  |
| 4    | CTS      | <b>Status of Clear To Send.</b><br>When MDCE bit is 1, this bit is the complement of CTS_ input. If Loop bit of UMCR is 1, this bit is equivalent to RTS bit of UMCR.<br>0: CTS_ pin is 1<br>1: CTS_ pin is 0  | R  |
| 3:1  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                           | R  |
| 0    | CCTS     | <b>Change status of CTS_.</b><br>When MDCE bit is 1, this bit indicates the state change on CTS_ pin.<br>0: No state change on CTS_ pin since last read of UMSR<br>1: A change occurs on the state of CTS_ pin | R  |

### 31.2.12 UART Scratchpad Register

This Scratchpad register is used as a scratch register for the programmer and has no effect on the UART.

|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |              |   |   |   |   |   |   |   |
|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------|--------------|---|---|---|---|---|---|---|
| <b>USPR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              | <b>0x01C</b> |   |   |   |   |   |   |   |
| <b>Bit</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Scratch Data |              |   |   |   |   |   |   |   |
| <b>RST</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              | ?            | ? | ? | ? | ? | ? | ? | ? |



### 31.2.13 Infrared Selection Register (ISR)

The ISR is used to configure the slow-infrared (SIR) interface that is provided in each UART to support two-way wireless communication using infrared transmission that conforms to the IrDA serial infrared specification 1.1. The maximum frequency is up to 115.2kbps.

|     |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  |       |      |       |        |        |   |   |   |
|-----|--|--|--|--|--|--|--|--|--|--|--|--|----------|--|--|-------|------|-------|--------|--------|---|---|---|
| ISR |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  | 0x020 |      |       |        |        |   |   |   |
| Bit |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  | 7     | 6    | 5     | 4      | 3      | 2 | 1 | 0 |
|     |  |  |  |  |  |  |  |  |  |  |  |  | Reserved |  |  | RDPL  | TDPL | XMODE | RCVEIR | XMITIR |   |   |   |
| RST |  |  |  |  |  |  |  |  |  |  |  |  |          |  |  | 0     | 0    | 0     | 0      | 0      | 0 | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                                                     | RW |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 7:5  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                            | R  |
| 4    | RDPL     | <b>Receive Data Polarity.</b><br>0: Slow-infrared (SIR) interface decoder takes positive pulses as zeros<br>1: SIR decoder takes negative pulses as zeros                                                                                                                                                                                                                       | W  |
| 3    | TDPL     | <b>Transmit Data Polarity.</b><br>0: SIR encoder generates a positive pulse for a data bit of zero<br>1: SIR encoder generates a negative pulse for a data bit of zero                                                                                                                                                                                                          | W  |
| 2    | XMODE    | <b>Transmit Pulse Width Mode.</b><br>Set when the transmit encoder needs to generate 1.6us pulses (that are 3/16 of a bit-time at 115.2 kbps).<br>Cleared when the transmit encoder needs to generate 3/16 of a bit-time wide according to current baud rate.<br>0: Transmit pulse width is 3/16 of a bit-time wide<br>1: Transmit pulse width is 1.6 us                        | W  |
| 1    | RCVEIR   | <b>Receiver SIR Enable.</b><br>This bit is used to select the signal from the RXD pin is processed by the IrDA decoder before it is fed to the UART (RCVEIR = 1) or bypass IrDA decoder and is fed directly to the UART (RCVEIR = 0).<br>0: Receiver is in UART mode<br>1: Receiver is in SIR mode                                                                              | W  |
| 0    | XMITIR   | <b>Transmitter SIR Enable.</b><br>This bit is used to select TXD output pin is processed by the IrDA encoder before it is fed to the device pin (XMITIR = 1) or bypass IrDA encoder and is fed directly to the device pin (XMITIR = 0).<br><br><b>NOTE:</b> disable infrared LED before XMITIR is set, otherwise a false start bit may occur.<br>0: Transmitter is in UART mode | W  |

|  |                               |  |
|--|-------------------------------|--|
|  | 1: Transmitter is in SIR mode |  |
|--|-------------------------------|--|

### 31.2.14 UART M Register (UMR)

|            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |          |  |  |   |  |  |  |  |  |   |   |   |   |   |   |   |   |
|------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------|----------|--|--|---|--|--|--|--|--|---|---|---|---|---|---|---|---|
| <b>UMR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x024</b> |          |  |  |   |  |  |  |  |  |   |   |   |   |   |   |   |   |
| <b>Bit</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |          |  |  |   |  |  |  |  |  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              | Reserved |  |  | M |  |  |  |  |  |   |   |   |   |   |   |   |   |
| <b>RST</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |          |  |  |   |  |  |  |  |  |   |   |   |   |   |   |   | 0 |

M is the value of UMR register.  
 It will take UART at least M cycles to transmit or receive one bit.  
 It will take UART at most M+1 cycles to transmit or receive one bit.

### 31.2.15 UART Add Cycle Register (UACR)

|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |          |  |  |    |  |  |  |  |  |    |    |   |   |   |   |   |   |   |   |   |   |
|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------|----------|--|--|----|--|--|--|--|--|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>UACR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x028</b> |          |  |  |    |  |  |  |  |  |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |          |  |  |    |  |  |  |  |  | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              | Reserved |  |  | AC |  |  |  |  |  |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |          |  |  |    |  |  |  |  |  |    |    |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 |

If nth bit of the register is 1, it will take UART M+1 cycles to transmit or receive the bit of date.  
 If the register is 12'h0, UART will receive or transmit a bit by M cycle(s).  
 If the register is 12'hfff, UART will receive or transmit a bit by M+1 cycle(s).  
 For the detail to see [For any frequency clock to use the Uart.](#)

### 31.2.16 UART RXFIFO Counter Register (URCR)

|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |          |  |  |      |  |  |  |  |  |   |   |   |   |   |   |   |   |
|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------|----------|--|--|------|--|--|--|--|--|---|---|---|---|---|---|---|---|
| <b>URCR</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x040</b> |          |  |  |      |  |  |  |  |  |   |   |   |   |   |   |   |   |
| <b>Bit</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |          |  |  |      |  |  |  |  |  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              | Reserved |  |  | RCNT |  |  |  |  |  |   |   |   |   |   |   |   |   |
| <b>RST</b>  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |              |          |  |  |      |  |  |  |  |  |   |   |   |   |   |   |   | 0 |

| Bits | Name | Description | RW |
|------|------|-------------|----|
|------|------|-------------|----|

|     |          |                                                                            |   |
|-----|----------|----------------------------------------------------------------------------|---|
| 7   | Reserved | Writing has no effect, read as zero.                                       | R |
| 6:0 | RCNT     | RXFIFO Counter. Indicates there are n data in RXFIFO when this field is n. | R |

### 31.2.17 UART TXFIFO Counter Register (UTCR)

|      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          |      |  |  |   |   |   |   |   |   |   |   |
|------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----------|------|--|--|---|---|---|---|---|---|---|---|
| UTCR |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x044    |      |  |  |   |   |   |   |   |   |   |   |
| Bit  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          |      |  |  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved | TCNT |  |  |   |   |   |   |   |   |   |   |
| RST  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |          |      |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                | RW |
|------|----------|----------------------------------------------------------------------------|----|
| 7    | Reserved | Writing has no effect, read as zero.                                       | R  |
| 6:0  | TCNT     | TXFIFO Counter. Indicates there are n data in TXFIFO when this field is n. | R  |

## 31.3 Operation

The following sections describe the UART operations that include flow of configuration, data transmission, data reception, and Infrared mode.

### 31.3.1 UART Configuration

Before UART starts to transfer data or changing transfer format, configuration must be done to define the transfer format. The sample flow is as the following:

In FIFO mode, set FME bit of UFCR to 1, reset receive and transmit FIFO, then initialize the UART as described below:

- 1 Clear UFCR.UME to 0.
- 2 Set value in UDLL/UDHR to generate the baud rate clock.
- 3 Set data format in ULCR.
- 4 If it is in FIFO MODE, set FME bit and other FIFO control in UFCR, reset receive and transmit FIFO, otherwise skip item 4.
- 5 Set each interrupt enable bit in UIER in interrupt-based transfer or set UFCR.DME in DMA-based transfer (DMA transfer is FIFO mode only), then set UFCR.UME.

### 31.3.2 Data Transmission

After configuration, UART is ready for data transfer. For data transmission, refer to the following procedure:

- 1 Read ULSR.TDRQ (interrupt disable) or wait for transmit data request interrupt (interrupt enable), if TDRQ = 1 or transmit data request interrupt generates, that means there is enough empty location in UTHR for new data.
- 2 If ULSR.TDRQ is 1 or get the transmit data request interrupt, write transmit data to UTHR to start transmission.
- 3 Do item 1 and item 2 if there are more data waiting for transmit.
- 4 After all necessary data are written to UTHR, wait ULSR.TEMP = 1, that means all data completely transmitted.
- 5 If it is necessary to send break, set ULCR.SBK and at least wait for 1-bit interval time to send a valid break, then clear ULCR.SBK.
- 6 Clear UME bit to finish UART transmission.

### 31.3.3 Data Reception

After configuration, UART is ready for data transfer. For data reception, refer to the following sample procedure:

- 1 Read ULSR.DRY (interrupt disable) or wait for receive data request interrupt (interrupt enable), if ULSR.DRY = 1 or receive data request interrupt generates, that means URBR has one data (non-FIFO mode) or data in URBR reaches the trigger value. (FIFO mode)
- 2 If ULSR.DRY = 1 or receive data request interrupt generates, then read ULSR.FIFOE or see if there is error interrupt, if FIFOE = 1, it means received data has receive error, then go to error handler, other wise go to item 3.
- 3 Read one received data in URBR (non-FIFO mode) or data equal to trigger value in URBR. (FIFO mode)
- 4 Check whether all data received: check whether ULSR.DRY = 0, in FIFO mode and interrupt is enabled, timeout interrupt may generate, when timeout interrupt generates, read URBR till ULSR.DRY = 0.
- 5 Clear UFCR.UME to end data reception when all data are received and ULSR.DRY = 0.

### 31.3.4 Receive Error Handling

A sample error handling flow is as the following:

- 1 If ULSR.FIFOE = 1, it means there is receive error in received data, then check what error it is.
- 2 If ULSR.OVER = 1, go to OVER error handling.
- 3 If ULSR.BI = 1, go to Break handling.
- 4 If ULSR.FMER = 1, go to Frame error handling.
- 5 If PARER = 1, go to PARER error handling.

### 31.3.5 Modem Transfer

When `UMCR.MDCE = 1`, modem control is enabled. Transfer flow can be stopped and restarted by software through `RTS_` and `CTS_` pin. When UART transmitter detects low level on `CTS_` pin, it stops transmission and `TxD` pin goes to mark state after finishing transmitting the current character until it detects `CTS_` pin goes back to high level. `RTS_` pin is output to receiving UART and its state can be controlled by setting `UMCR.RTS` bit, that is, setting `UMCR.RTS` to 1, `RTS_` pin is low level output that means UART is ready to receive data, on the contrary, it means UART currently can't receive more data.

### 31.3.6 DMA Transfer

UART can operate in DMA-based (`UFCR.DME = 1`, FIFO mode only), that is, dma request initiated by UART takes the place of interrupt request and transmission/reception is carried out using DMA instead of CPU. Be sure that software guarantee to disable transmit and receive interrupt except timeout and error interrupts.

During DMA transfer, if an interrupt occurs, software must first read the `ULSR` to see if an error interrupt exists, then check the `UIIR` for the source of the interrupt and if DMA channel is already halt because of the error indicator from UART, then disable DMA channel and read out all the error data from receive FIFO. Software re-set and re-enable DMA and data transfer by DMA will re-start.

### 31.3.7 Slow IrDA Asynchronous Interface

Each UART supports slow infra-red (SIR) transmission and reception by setting `ISR.XMITIR` and `ISR.RCVEIR` to 1 (make sure the two bits are not set to 1 at the same time because SIR can't operate full-duplex). According to the IrDA 1.1, data rate is limited at a maximum value of 115.2Kbps.

In SIR transmit mode, the transmit pulse comes out at a rate of 3/16 (when the transmit data bit is zero); in SIR receive mode, the receiver must detect the 3/16 pulsed period to recognize a zero value (an active high or low pulse is demodulation to 0, and no pulse is demodulation to 1).

Compared to normal UART, there are some limitations to SIR, that is, each character is fixed to 8-bit data width, no parity and 1 stop bit and modem function is ignored. The IrDA 1.1 specifies a minimum 10ms latency after an optical node ceases transmitting before its receiver recovers its receiving function and software must guarantee this delay.

In the IrDA 1.1 specification, communication must start up at the rate of 9600bps, but then allows the link to negotiate higher (or lower) data rates if supported by both ends. However, the communication rate will not automatically change. Change, if necessary, is performed by software.

### 31.3.8 For any frequency clock to use the UART

**NOTE:** if you don't set M register and UACR the UART work at normal mode with the specified frequencies. To use other frequency you should to set M register and UACR to right value.

### 1 The Improving

Following changes are made:

- a One bit is composed by M CLK<sub>BR</sub> cycles, which can be 4~1024.
- b Some extra CLK<sub>BR</sub> cycles can be inserted in some bits in one frame, so that like M has fraction.

For instance:

$$CLK_{BR} = CLK_{DEV} / N \quad N = 1, 2, \dots$$

$$CLK_{BR} = CLK_{DEV} = 4MHz$$

$$\text{Band rate} = 460800$$

In accurate

$$M_a = 8.681$$

We take

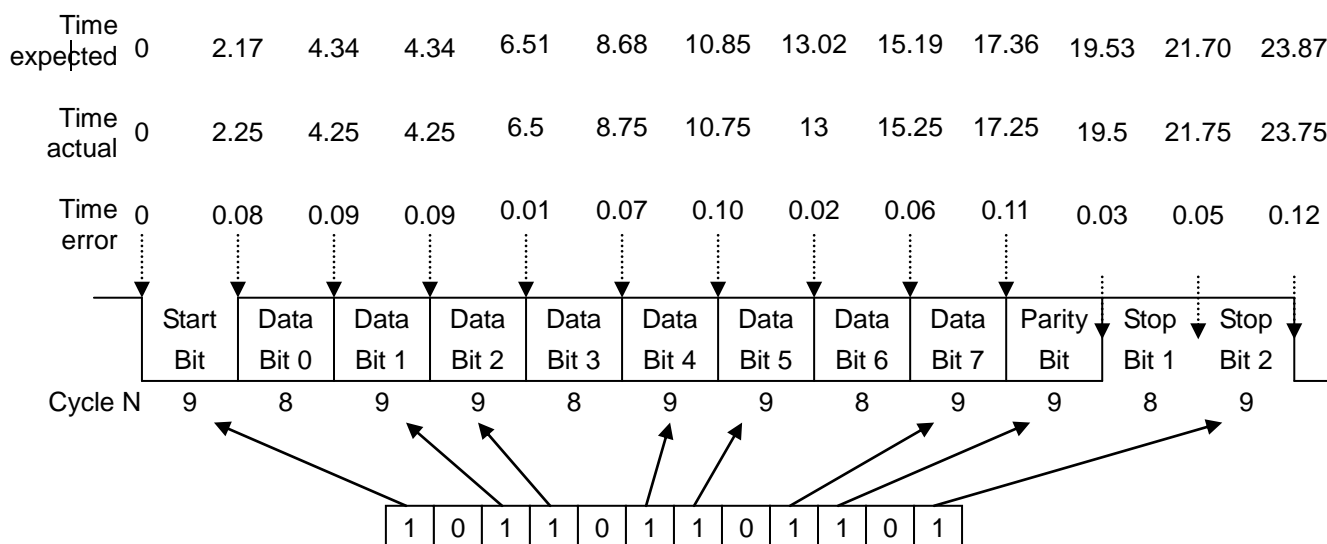
$$M = 8, \text{ with } 8 \text{ extra cycles in every frame}$$

A 12-bit register is used to indicate where to insert the extra cycles.

The first line is the time expected

The second line is the time actual

The third line is the time error



For transmission, in theory, the biggest error is half of CLK<sub>BR</sub> cycle, which is 0.125us here.

### 2 To set UMR register

$$CLK_{BR} = CLK_{DEV} / N$$

$M_a = CLK_{BR} / \text{band rate}$

M is modum of  $M_a$ .

Write M to Mregister.

Considering the power and the robust quality, for M form 6 to 32 is you better select by set the UDLR.

The max error

$$\frac{0.5 / CLK_{BR}}{M_a / CLK_{BR}} = 0.5 / M_a < 0.5 / M$$

|                  |       |       |        |       |       |
|------------------|-------|-------|--------|-------|-------|
| M                | 4     | 8     | 16     | 32    | 64    |
| error/ $W_{bit}$ | 12.5% | 6.25% | 3.125% | 1.56% | 0.78% |

### 3 To set UACR value

For each bit of it means:

0: means not to add additional cycle to the bit that UART is prepare to transmit or receive, in another word, you will to use M cycles to transmit or receive the bit

1: means to add additional cycle to the bit that UART is prepare to transmit or receive, in another word, you will to use M+1 cycles to transmit or receive the bit

To set UACR value you must ensure that the max error of each bit should be less than  $0.5P_{BR}$ .

For example:  $M_a - M = 0.15$ ;  $M + 1 - M_a = 0.85$ ;

Write 8 to UMR,

Write 0x408 to UACR

cycle/bit : M, M, M, M+1, M, M, M, M, M, M, M+1, M

UACR : 0 0 0 1 0 0 0 0 0 0 1 0

## 32 MMC/SD CE-ATA Controller

### 32.1 Overview

The Multi Media Card (MMC) is a universal low cost mass storage and communication media that is designed to cover a wide area of applications such as electronic toys, organizers, PDAs, smart phones, and so on.

The Secure Digital (SD) card is an evolution of MMC, It is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in newly emerging audio and video consumer electronic devices. The physical form factor, pin assignment, and data transfer protocol are forward compatible with the MultiMediaCard with some additions. An SD card can be categorized as SD memory or SD I/O card, commonly known as SDIO. A memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard and is faster and capable of higher memory capacity. The SDIO card provides high-speed data I/O with low-power consumption for mobile electronic devices.

For CE-ATA detail protocol , please referred to [WWW.CE-ATA.ORG](http://WWW.CE-ATA.ORG).

### 32.2 Features

- Fully compatible with the MMC System Specification version 4.2
- Support SD Specification 3.0
- Support SD I/O Specification 1.0 with 1 command channel and 4 data channels
- Consumer Electronics Advanced Transport Architecture (CE-ATA – version 1.1)
- Maximum data rate is 50MBps
- Support MMC data width 1bit ,4bit and 8bit
- Built-in programmable frequency divider for MMC/SD bus
- Built-in Special Descriptor DMA
- Maskable hardware interrupt for SDIO interrupt, internal status and FIFO status
- 128 x 32 built-in data FIFO
- Multi-SD function support including multiple I/O and combined I/O and memory
- IRQ supported enable card to interrupt MMC/SD controller
- Single or multi block access to the card including erase operation
- Stream access to the MMC card
- Supports SDIO read wait, interrupt detection during 1-bit or 4-bit access
- Supports CE-ATA digital protocol commands
- Support Command Completion Signal and interrupt to CPU
- Command Completion Signal disable feature
- The maximum block length is 4096bytes

### 32.3 Pins Description

- MSC\_CLK, output, host to card clock signal.
- MSC\_CMD, inout, bidirectional command/response signal.
- MSC\_DAT[7:0], inout, bidirectional data bus.



### 32.4 Block Diagram

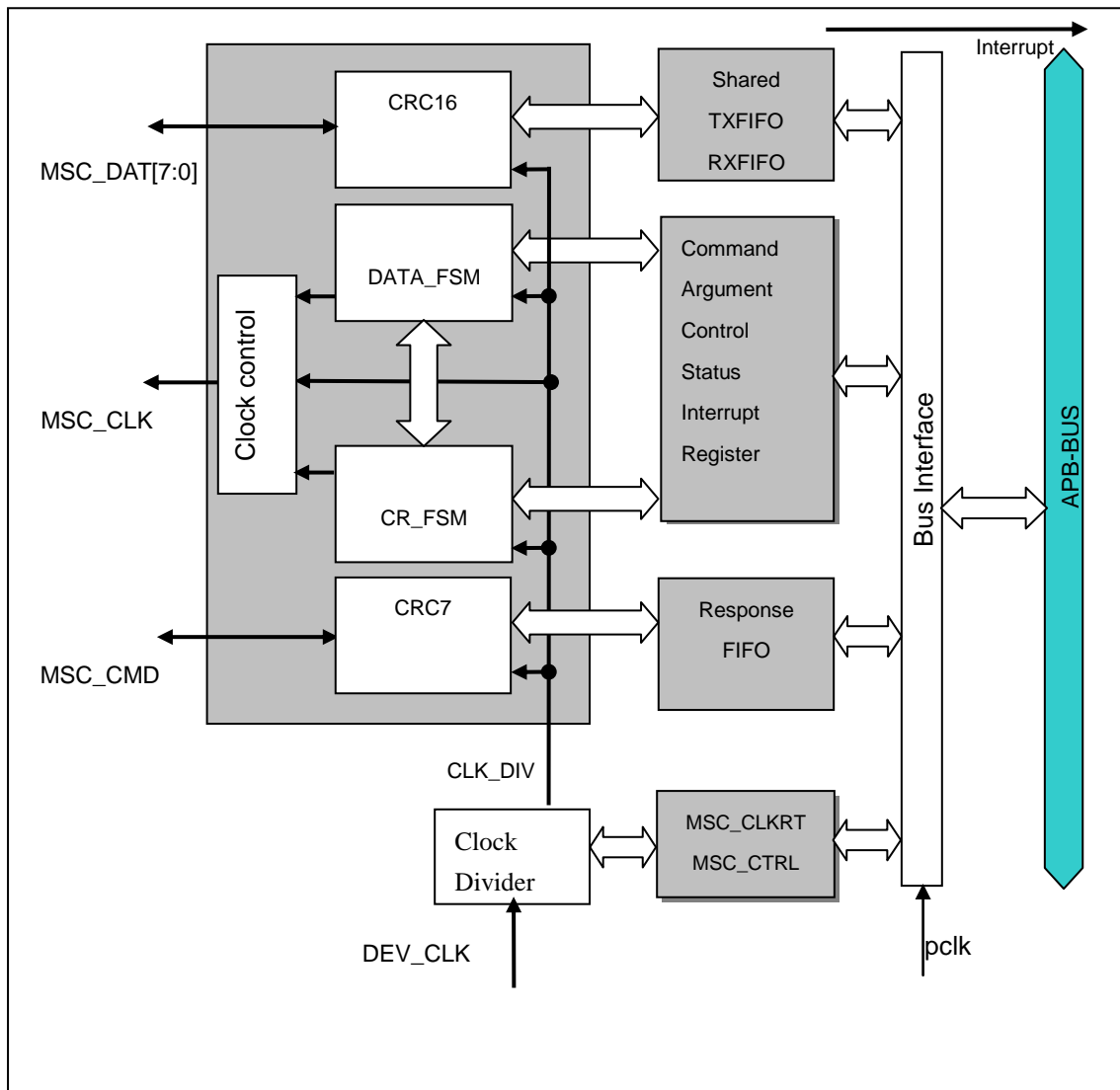


Figure 32-1 MMC/SD Controller Block Diagram

### 32.5 MMC/SD Controller Signal I/O Description

MSC and the card communication over the CMD and DATA line is based on command and data bit streams which are initiated by a start bit and terminated by a stop bit.

**Command:** a command is a token, which starts an operation. A command is sent from MSC either to a single card (addressed command) or to all connected cards (broadcast command). A command is transferred serially on the CMD line. Each command token is preceded by a start bit ('0') and succeeded by an end bit ('1'). The total length is 48 bits and protected by CRC bits.

**Table 32-1 Command Token Format**

| Bit position | 47        | 46               | [45 : 40]     | [39 : 8] | [7 : 1] | 0       |
|--------------|-----------|------------------|---------------|----------|---------|---------|
| Width (bits) | 1         | 1                | 6             | 32       | 7       | 1       |
| Value        | 0         | 1                | X             | X        | x       | 1       |
| Description  | Start bit | Transmission bit | Command index | argument | CRC7    | End bit |

**Response:** a response is a token which is sent from an addressed card, or (synchronously) from all connected cards, to MSC as an answer to a previously received command. A response is transferred serially on the CMD line. Response tokens have various coding schemes depending on their content.

**Data:** data can be transferred from the card to MSC or vice versa. Data is transferred via the data line. Data transfers to/from the SD Memory Card are done in blocks. Data blocks are always succeeded by CRC bits. Single and multiple block operations are defined. Note that the Multiple Block operation mode is better for faster write operation. A multiple block transmission is terminated when a stop command follows on the CMD line. Data transfer can be configured by the MSC to use single or multiple data lines.

**Table 32-2 MMC/SD Data Token Format**

| Description | Start bit | Data | CRC16  | End bit |
|-------------|-----------|------|--------|---------|
| Stream Data | 0         | X    | no CRC | 1       |
| Block Data  | 0         | X    | X      | 1       |

## 32.6 Register Description

The MMC-SD-CE\_ATA controller is controlled by a set of registers that the application configures before every operation. The Table 32-3 lists all the MSC registers.

**Table 32-3 MMC/SD Controller Registers Map**

| Name       | RW | Reset Value | Address Offset | Access Size |
|------------|----|-------------|----------------|-------------|
| MSC_CTRL   | W  | 0x0000      | 0x00           | 16          |
| MSC_STAT   | R  | 0x00000040  | 0x04           | 32          |
| MSC_CLKRT  | RW | 0x0000      | 0x08           | 16          |
| MSC_CMDAT  | RW | 0x00000000  | 0x0c           | 32          |
| MSC_RESTO  | RW | 0x100       | 0x10           | 16          |
| MSC_RDTO   | RW | 0xFFFFFFFF  | 0x14           | 32          |
| MSC_BLKLEN | RW | 0x0000      | 0x18           | 16          |
| MSC_NOB    | RW | 0x0000      | 0x1c           | 16          |

|             |    |            |      |    |
|-------------|----|------------|------|----|
| MSC_SNOB    | R  | 0x????     | 0x20 | 16 |
| MSC_IMASK   | RW | 0xFFFFFFFF | 0x24 | 32 |
| MSC_IFLG    | RW | 0x2000     | 0x28 | 32 |
| MSC_CMD     | RW | 0x00       | 0x2c | 8  |
| MSC_ARG     | RW | 0x00000000 | 0x30 | 32 |
| MSC_RES     | R  | 0x????     | 0x34 | 16 |
| MSC_RXFIFO  | R  | 0x???????? | 0x38 | 32 |
| MSC_TXFIFO  | W  | 0x???????? | 0x3c | 32 |
| MSC_LPM     | RW | 0x00000000 | 0x40 | 32 |
| MSC_DMACH   | RW | 0x00000000 | 0x44 | 32 |
| MSC_DMACHA  | RW | 0x00000000 | 0x48 | 32 |
| MSC_DMACHB  | RW | 0x00000000 | 0x4c | 32 |
| MSC_DMALEN  | RW | 0x00000000 | 0x50 | 32 |
| MSC_DMACHMD | RW | 0x00000000 | 0x54 | 32 |
| MSC_CTRL2   | RW | 0x00800000 | 0x58 | 32 |
| MSC_RTCNT   | R  | 0x00000000 | 0x5c | 32 |

Note:

There are three MSC controllers in the chip, named MSC0, MSC1 and MSC2.

The base address of MSC0 controller is 0x13450000.

The base address of MSC1 controller is 0x13460000.

The base address of MSC2 controller is 0x13470000.

### 32.6.1 MSC Control Register (MSC\_CTRL)

| MSC_CTRL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |           |              |          |    |    | 0x00 |   |   |               |               |                |               |       |          |            |   |   |   |
|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-----------|--------------|----------|----|----|------|---|---|---------------|---------------|----------------|---------------|-------|----------|------------|---|---|---|
| Bit      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15        | 14           | 13       | 12 | 11 | 10   | 9 | 8 | 7             | 6             | 5              | 4             | 3     | 2        | 1          | 0 |   |   |
|          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | SEND_CCSD | SEND_AD_CCSD | Reserved |    |    |      |   |   | EXIT_MULTIPLE | EXIT_TRANSFER | START_READWAIT | STOP_READWAIT | RESET | START_OP | CLOCK_CTRL |   |   |   |
| RST      |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0         | 0            | 0        | 0  | 0  | 0    | 0 | 0 | 0             | 0             | 0              | 0             | 0     | 0        | 0          | 0 | 0 | 0 |

| Bits | Name      | Description                                                                                                                                                                                                                                                                   | RW |
|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 15   | SEND_CCSD | 0: clear bit<br>1: Send Command Completion Signal Disable (CCSD) to CE_ATA device<br>when set, host sends CCSD to CE_ATA device. Software set the bit only if current command is expecting CCS and interrupts are enabled in CE_ATA devices. Once the CCSD pattern is sent to | W  |

|      |                |                                                                                                                                                                                                                                                                                                                                                                                         |   |
|------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|      |                | device, host automatically clears the SEND_CCSD bit.                                                                                                                                                                                                                                                                                                                                    |   |
| 14   | SEND_AS_CCSD   | <p>0: clear bit</p> <p>1: send internally generated stop after sending CCSD to CE_ATA device</p> <p>When set, host automatically sends internally-generated STOP command(CMD12) to CE_ATA device.</p> <p>After sending CMD12, MSC_STAT. AUTO_CMD12_DONE is set and generates interrupt to CPU.</p> <p>After sending the CCSD, controller automatically clears the SEND_AS_CCSD bit.</p> | W |
| 13:8 | Reserved       | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                    | R |
| 7    | EXIT_MULTIPLE  | <p>If CMD12 or CMD52 (I/O abort) is to be sent to terminate multiple block read/write in advance, set this bit to 1.</p> <p>0: No effect</p> <p>1: Exit from multiple block read/write</p>                                                                                                                                                                                              | W |
| 6    | EXIT_TRANSFER  | <p><b>Only used for SDIO suspend/resume and MMC stream read.</b></p> <p>For SDIO, after suspend is accepted, set this bit with 1.</p> <p>For MMC, after the data of the expected number are received, set this bit with 1.</p> <p>0: No effect</p> <p>1: Exit from multiple block read/write after suspend is accepted, or exit from stream read</p>                                    | W |
| 5    | START_READWAIT | <p>Only used for SDIO ReadWait. Start the ReadWait cycle.</p> <p>0: No effect</p> <p>1: Start ReadWait</p>                                                                                                                                                                                                                                                                              | W |
| 4    | STOP_READWAIT  | <p>Only used for SDIO ReadWait. Stop the ReadWait cycle.</p> <p>0: No effect</p> <p>1: Start ReadWait</p>                                                                                                                                                                                                                                                                               | W |
| 3    | RESET          | <p>Reset the MSC controller.</p> <p>0: No effect</p> <p>1: Reset the MSC controller</p>                                                                                                                                                                                                                                                                                                 | W |
| 2    | START_OP       | <p>This bit is used to start the new operation. When starting the clock, this bit can be 1. When stopping the clock, this bit can only be 0.</p> <p>0: Do nothing</p> <p>1: Start the new operation</p>                                                                                                                                                                                 | W |
| 1:0  | CLOCK_CTRL     | <p>These bits are used to start or stop clock.</p> <p>00: Do nothing</p> <p>01: Stop MMC/SD clock</p> <p>10: Start MMC/SD clock</p> <p>11: Reserved</p>                                                                                                                                                                                                                                 | W |

### 32.6.2 MSC Control 2 Register (MSC\_CTRL2)

| MSC_CTRL2 |          |    |    |     |    |    |    |    |        |          |    |    |    |    |    | 0x58 |    |    |    |    |    |       |     |     |   |   |   |   |   |   |   |   |
|-----------|----------|----|----|-----|----|----|----|----|--------|----------|----|----|----|----|----|------|----|----|----|----|----|-------|-----|-----|---|---|---|---|---|---|---|---|
| Bit       | 31       | 30 | 29 | 28  | 27 | 26 | 25 | 24 | 23     | 22       | 21 | 20 | 19 | 18 | 17 | 16   | 15 | 14 | 13 | 12 | 11 | 10    | 9   | 8   | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|           | Reserved |    |    | PIP |    |    |    |    | RST_EN | Reserved |    |    |    |    |    |      |    |    |    |    |    | STPRM | SVC | SMS |   |   |   |   |   |   |   |   |
| RST       | 0        | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 1      | 0        | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0     | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                            | RW |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:29 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                   | R  |
| 28:24 | PIP      | Pin Level Interrupt Polarity<br>1: Interrupt can be triggered when pin level is high<br>0: Interrupt can be triggered when pin level is low                                                                                                                                                                                                                                                                            | RW |
| 23    | RST_EN   | Control the level of pin RST_n to reset the card<br>0: Disable reset or do nothing<br>1: Enable reset to card                                                                                                                                                                                                                                                                                                          | RW |
| 22:5  | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                                                   | R  |
| 4     | STPRM    | Stop Read Operation Mode Selection<br>0: Host can stop read operation during data transfer<br>1: Host only can stop read operation during block gap                                                                                                                                                                                                                                                                    | RW |
| 3     | SVC      | Signal Voltage Change Control<br>1: Start signal voltage change from 3.3V to 1.8V. Hardware will stop MSC_CLK at low.<br>0: Stop signal voltage change from 3.3V to 1.8V. Hardware will start to provide MSC_CLK at 1.8 V.<br>1.8V regulator output should be stable within 5ms when changing voltage from 3.3V to 1.8V.<br>3.3V regulator output should be stable within 5ms when changing voltage from 1.8V to 3.3V. | RW |
| 2:0   | SMS      | Speed Mode Selection<br>000: Default speed<br>001: High speed<br>010: SDR12<br>011: SDR25<br>100: SDR50<br>Others: reserved                                                                                                                                                                                                                                                                                            | RW |

### 32.6.3 MSC Status Register (MSC\_STAT)

| MSC_STAT |                 |                 |     |           |    |    |    |          |    |    |    |     |     |     |     | 0x04   |              |                 |          |                |             |                 |             |        |                |                 |             |                |                 |              |              |   |
|----------|-----------------|-----------------|-----|-----------|----|----|----|----------|----|----|----|-----|-----|-----|-----|--------|--------------|-----------------|----------|----------------|-------------|-----------------|-------------|--------|----------------|-----------------|-------------|----------------|-----------------|--------------|--------------|---|
| Bit      | 31              | 30              | 29  | 28        | 27 | 26 | 25 | 24       | 23 | 22 | 21 | 20  | 19  | 18  | 17  | 16     | 15           | 14              | 13       | 12             | 11          | 10              | 9           | 8      | 7              | 6               | 5           | 4              | 3               | 2            | 1            | 0 |
|          | AUTO_CMD12_DONE | AUTO_CMD23_DONE | SVS | PIN_LEVEL |    |    |    | Reserved |    |    |    | BCE | BDE | BAE | BAR | DMAEND | IS_RESETTING | SDIO_INT_ACTIVE | PRG_DONE | DATA_TRAN_DONE | END_CMD_RES | DATA_FIFO_AFULL | IS_READWAIT | CLK_EN | DATA_FIFO_FULL | DATA_FIFO_EMPTY | CRC_RES_ERR | CRC_READ_ERROR | CRC_WRITE_ERROR | TIME_OUT_RES | TIME_OUTREAD |   |
| RST      | 0               | 0               | 0   | 0         | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0      | 0            | 0               | 0        | 0              | 0           | 0               | 0           | 0      | 0              | 1               | 0           | 0              | 0               | 0            | 0            | 0 |

| Bits  | Name            | Description                                                                                                                                                          | RW |
|-------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | AUTO_CMD12_DONE | Indicates that the stop command (CMD12) that is internally generated by controller has finished.                                                                     | R  |
| 30    | AUTO_CMD23_DONE | Indicates that the auto command CMD23 is finished<br>0: Not finished<br>1: Finished                                                                                  | R  |
| 29    | SVS             | Signal Voltage Change Status<br>1: MSC_CLK starts stopping at low<br>0: MSC_CLK has no change of stopping at low                                                     | R  |
| 28:24 | PIN_LEVEL       | MSC Interface Pin Level<br>PIN_LEV[4:0] indicates the pin level of MSC_CMD, MSC_DAT[3:0] separately.                                                                 | R  |
| 23:21 | Reserved        | Writing has no effect, read as zero.                                                                                                                                 | R  |
| 20    | BCE             | Boot CRC error.<br>0: No boot CRC error occurs.<br>1: Boot CRC is not correct.                                                                                       | R  |
| 19    | BDE             | Boot data end.<br>0: No boot data or boot data is not finished.<br>1: Boot data is received completely.                                                              | R  |
| 18    | BAE             | Boot acknowledge is error.<br>0: No boot acknowledge is received or boot acknowledge is correct (decided by BAR)<br>1: The received boot acknowledge is not correct. | R  |
| 17    | BAR             | Boot acknowledge received.<br>0: No boot acknowledge received.<br>1: Boot acknowledge received.                                                                      | R  |
| 16    | DMAEND          | Indicates that the DMA has finished the current transfer.                                                                                                            | R  |
| 15    | IS_RESETTING    | MSC is resetting after power up or MSC_CTRL[RESET] is written with 1.                                                                                                | R  |

|     |                 |                                                                                                                                                                                                                      |   |
|-----|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|     |                 | 0: Reset has been finished<br>1: Reset has not been finished                                                                                                                                                         |   |
| 14  | SDIO_INT_ACTIVE | Indicates whether an interrupt is detected at the SD I/O card. A separate acknowledge command to the card is required to clear this interrupt.<br>0: No interrupt detected<br>1: The interrupt from SDIO is detected | R |
| 13  | PRG_DONE        | Indicates whether card has finished programming.<br>0: not finished<br>1: finished                                                                                                                                   | R |
| 12  | DATA_TRAN_DONE  | Indicates whether data transmission to card has completed.<br>0: not completed<br>1: completed                                                                                                                       | R |
| 11  | END_CMD_RES     | Indicates whether command and response/no-response sequence have been completed<br>0: not completed<br>1: completed                                                                                                  | R |
| 10  | DATA_FIFO_AFULL | Indicates whether the data FIFO is almost full.<br>0: The number of words in FIFO is less than 127.<br>1: The number of words in FIFO is equal to or greater than 127                                                | R |
| 9   | IS_READWAIT     | Indicates whether SDIO card has entered ReadWait State.<br>0: Card has not entered ReadWait<br>1: Card has entered ReadWait                                                                                          | R |
| 8   | CLK_EN          | Clock enabled.<br>0: Clock is off<br>1: Clock is on                                                                                                                                                                  | R |
| 7   | DATA_FIFO_FULL  | Indicates whether the data FIFO is full.<br>0: Data FIFO is not full<br>1: Data FIFO is full                                                                                                                         | R |
| 6   | DATA_FIFO_EMPTY | Indicates whether data FIFO is empty.<br>0: Data FIFO is not empty<br>1: Data FIFO is empty                                                                                                                          | R |
| 5   | CRC_RES_ERR     | Response CRC error.<br>0: No error on the response CRC<br>1: CRC error occurred on the response                                                                                                                      | R |
| 4   | CRC_READ_ERROR  | Read CRC error.<br>0: No error on received data<br>1: CRC error occurred on received data                                                                                                                            | R |
| 3:2 | CRC_WRITE_ERROR | Write CRC error.<br>00: No error on transmission of data                                                                                                                                                             | R |

|   |               |                                                                                                    |   |
|---|---------------|----------------------------------------------------------------------------------------------------|---|
|   |               | 01: Card observed erroneous transmission of data<br>10: No CRC status is sent back<br>11: Reserved |   |
| 1 | TIME_OUT_RES  | Response Time Out.<br>0: Card response has not timed out<br>1: Card response has time out          | R |
| 0 | TIME_OUT_READ | Read time out.<br>0: Card read data has not timed out<br>1: Card read data has timed out           | R |

### 32.6.4 MSC Clock Rate Register (MSC\_CLKRT)

The MSC\_CLKRT register specifies the frequency division of the MMC/SD bus clock. The software is responsible for setting this register.

|                  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |    |    |    |    |    |   |   |   |   |          |   |   |   |   |   |
|------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|----|----|----|----|----|---|---|---|---|----------|---|---|---|---|---|
| <b>MSC_CLKRT</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x08</b> |    |    |    |    |    |   |   |   |   |          |   |   |   |   |   |
| <b>Bit</b>       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5        | 4 | 3 | 2 | 1 | 0 |
|                  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Reserved    |    |    |    |    |    |   |   |   |   | CLK_RATE |   |   |   |   |   |
| <b>RST</b>       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0        | 0 | 0 | 0 | 0 | 0 |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                  | RW |
|------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 15:3 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                         | R  |
| 2:0  | CLK_RATE | Clock rate.<br>000: DEV_CLK<br>001: 1/2 of DEV_CLK<br>010: 1/4 of DEV_CLK<br>011: 1/8 of DEV_CLK<br>100: 1/16 of DEV_CLK<br>101: 1/32 of DEV_CLK<br>110: 1/64 of DEV_CLK<br>111: 1/128 of DEV_CLK<br>This field must be set to 0 when the controller works during normal writing or reading. | RW |

### 32.6.5 MSC Command and Data Control Register (MSC\_CMDAT)

|                  |             |
|------------------|-------------|
| <b>MSC_CMDAT</b> | <b>0x0C</b> |
|------------------|-------------|



|     |              |            |          |          |          |              |           |          |    |    |    |    |    |            |           |            |      |    |      |    |          |           |   |          |      |      |              |            |         |                 |   |   |
|-----|--------------|------------|----------|----------|----------|--------------|-----------|----------|----|----|----|----|----|------------|-----------|------------|------|----|------|----|----------|-----------|---|----------|------|------|--------------|------------|---------|-----------------|---|---|
| Bit | 31           | 30         | 29       | 28       | 27       | 26           | 25        | 24       | 23 | 22 | 21 | 20 | 19 | 18         | 17        | 16         | 15   | 14 | 13   | 12 | 11       | 10        | 9 | 8        | 7    | 6    | 5            | 4          | 3       | 2               | 1 | 0 |
|     | CCS_EXPECTED | READ_CEATA | Reserved | DIS_BOOT | ENB_BOOT | EXP_BOOT_ACK | BOOT_MODE | Reserved |    |    |    |    |    | AUTO_CMD23 | SDIO_PDRT | AUTO_CMD12 | RTRG |    | TTRG |    | IO_ABORT | BUS_WIDTH |   | Reserved | INIT | BUSY | STREAM_BLOCK | WRITE_READ | DATA_EN | RESPONSE_FORMAT |   |   |
| RST | 0            | 0          | 0        | 0        | 0        | 0            | 0         | 0        | 0  | 0  | 0  | 0  | 0  | 0          | 0         | 0          | 0    | 0  | 0    | 0  | 0        | 0         | 0 | 0        | 0    | 0    | 0            | 0          | 0       | 0               | 0 | 0 |

| Bits  | Name         | Description                                                                                                                                                                                                                                                                                                                                                                                       | RW |
|-------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31    | CCS_EXPECTED | 0: interrupts are not enabled in CE-ATA device, or commands does not expect CCS from device<br>1: interrupts are enabled in CE_ATA device, or RW_BLK command expects command completion signal from device<br>If the command expects Command Completion Signal (CCS) from the device, the software should set the control bit. It is auto cleared 0 by hardware.                                  | RW |
| 30    | READ_CEATA   | 0: host is not performing read access (RW_BLK or RW_REG) towards CE_ATA device<br>1: host id performing read access (RW_BLK or RW_REG) towards CE_ATA device<br>Software should set the bit to indicate that CE_ATA device is being accessed for read transfer. The bit is used to disable read data timeout indication while performing CE_ATA read transfers. It is auto cleared 0 by hardware. | RW |
| 29:28 | Reserved     | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                              | R  |
| 27    | DIS_BOOT     | 0: Do nothing<br>1: Stop boot operation<br>DIS_BOOT should not be used with ENA_BOOT at the same time.                                                                                                                                                                                                                                                                                            | RW |
| 26    | ENB_BOOT     | This bit is used for Mandatory boot operation mode only.<br>0: Do nothing<br>1: Start Mandatory boot                                                                                                                                                                                                                                                                                              | RW |
| 25    | EXP_BOOT_ACK | Whether boot acknowledge pattern is expected or not<br>0: Boot acknowledge pattern is not expected<br>1: Boot acknowledge pattern is expected                                                                                                                                                                                                                                                     | RW |
| 24    | BOOT_MODE    | Boot mode operation selection.<br>0: Mandatory boot operation<br>1: Alternative boot operation                                                                                                                                                                                                                                                                                                    | RW |
| 23:19 | Reserved     | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                              | R  |
| 18    | AUTO_CMD23   | This field controls use of auto command functions.                                                                                                                                                                                                                                                                                                                                                | RW |

|       |            |                                                                                                                                                                                                                                                                                                                             |    |
|-------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|       |            | It can be used only when the card supports CMD23.<br>0: Auto CMD23 is disabled<br>1: MSC controller will issue CMD23 automatically before issuing a command specified in MSC_CMD register.<br>when stop command has finished, it is auto cleared 0 by hardware.                                                             |    |
| 17    | SDIO_PRDT  | Determine whether SDIO interrupt is 2 cycle or extend more cycle when data block last is transferred.<br>0: more cycle (like single block)<br>1: exact 2 cycle                                                                                                                                                              | RW |
| 16    | AUTO_CMD12 | This field controls use of auto command functions.<br>0: Auto CMD12 is disabled<br>1: MSC controller will issue CMD12 automatically after the last data block transfer is finished.<br>when stop command has finished, it is auto cleared 0 by hardware.                                                                    | RW |
| 15:14 | RTRG       | Receive FIFO Trigger Value Select.<br>These bits set the receive FIFO half-empty threshold value, when the number of transmit FIFO $\geq$ threshold value , RXFIFO_RD_REQ will be set to 1.<br>00: more than or equal to 16<br>01: more than or equal to 32<br>10: more than or equal to 64<br>11: more than or equal to 96 | RW |
| 13:12 | TTRG       | Transmit FIFO Trigger Value Select.<br>These bits set the transmit FIFO half-empty threshold value, when the number of transmit FIFO $<$ threshold value , TXFIFO_WR_REQ will be set to 1.<br>00: less than 16<br>01: less than 32<br>10: less than 64<br>11: less than 96                                                  | RW |
| 11    | IO_ABORT   | Specifies the current command is used to abort data transfer.<br>0: Nothing<br>1: The current command is used to abort transfer<br>It is auto cleared 0 by hardware.                                                                                                                                                        | WR |
| 10:9  | BUS_WIDTH  | Specifies the width of the data bus.<br>00: 1-bit<br>01: Reserved<br>10: 4-bit<br>11: 8bit                                                                                                                                                                                                                                  | WR |
| 8     | Reserved   | Writing has no effect.                                                                                                                                                                                                                                                                                                      | R  |

|     |                 |                                                                                                                                                                                                                                    |    |
|-----|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|     |                 | <b>Read as zero.</b>                                                                                                                                                                                                               |    |
| 7   | INIT            | 80 initialization clocks.<br>0: Issues command directly<br>1: Issues 80 clocks before command                                                                                                                                      | W  |
| 6   | BUSY            | Specifies whether a busy signal is expected after the current command. This bit is for no data command/response transactions only.<br>0: Not expect a busy signal<br>1: Expects a busy signal. If the response is R1b, then set it | WR |
| 5   | STREAM_BLOCK    | Stream mode.<br>0: Data transfer of the current command sequence is not in stream mode<br>1: Data transfer of the current command sequence is in stream mode                                                                       | WR |
| 4   | WRITE_READ      | Data Transfer Direction Selection.<br>0: Read (from card to host)<br>1: Write (from host to card)                                                                                                                                  | RW |
| 3   | DATA_EN         | Specifies whether the current command includes data transfer or not.<br>It is also used to reset RX_FIFO and TX_FIFO.<br>0: Current command without data transfer<br>1: Current command with data transfer                         | RW |
| 2:0 | RESPONSE_FORMAT | Response Type Selection.<br>000: No response<br>001: Format R1 and R1b<br>010: Format R2<br>011: Format R3<br>100: Format R4<br>101: Format R5<br>110: Format R6<br>111: Format R7                                                 | RW |

### 32.6.6 MSC Response Time Out Register (MSC\_RESTO)

|                  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|--------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| <b>MSC_RESTO</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x10</b> |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>       |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15          | 14     | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|                  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             | RES_TO |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>       |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0           | 0      | 0  | 0  | 0  | 0  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name   | Description                                                                                                                                             | RW |
|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 15:0 | RES_TO | Specifies the maximum number of MSC_CLK clock cycles between the end bit of the command and the response from the SD card.<br>The default value is 128. | RW |

### 32.6.7 MSC Read Time Out Register (MSC\_RDTO)

| MSC_RDTO |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x14 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|----------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| Bit      | 31      | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16   | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|          | READ_TO |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST      | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1    | 1  | 1  | 1  | 1  | 1  | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bits | Name    | Description                                                                                                                                                            | RW |
|------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | READ_TO | Specifies the maximum number of clocks between the command and when the MMC/SD host controller turns on the time-out error for the received data. The unit is MSC_CLK. | RW |

### 32.6.8 MSC Block Size Register (MSC\_BLKLEN)

| MSC_BLKLEN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x18 |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |  |
|------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|--|
| Bit        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |  |
|            |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      | BLK_LEN |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |  |
| RST        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      | 0       | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

| Bits | Name    | Description                                                                                                                                | RW |
|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------|----|
| 15:0 | BLK_LEN | Specifies the number of bytes in a block, and is normally set to 0x200 for MMC/SD data transactions. The value Specified in the cards CSD. | RW |

### 32.6.9 MSC Block Count Register (MSC\_NOB)

| MSC_NOB |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x1C |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |  |
|---------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|--|
| Bit     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |  |
|         |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      | NOB |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |  |
| RST     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      | 0   | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |

| Bits | Name | Description                                                                    | RW |
|------|------|--------------------------------------------------------------------------------|----|
| 15:0 | NOB  | Specifies the number of blocks in a data transfer. One block is a possibility. | RW |

### 32.6.10 MSC Successfully-transferred Blocks Count Register (MSC\_SNOB)

In block mode, the MSC\_SNOB register records the number of successfully transferred blocks. If the last block has CRC error, this register also summaries it. It is used to query blocks for multiple block transfer.

| MSC_SNOB |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x20 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Bit      |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|          | MSC_SNOB |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST      |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ?    | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? |   |

| Bits | Name     | Description                                                                          | RW |
|------|----------|--------------------------------------------------------------------------------------|----|
| 15:0 | MSC_SNOB | Specify the number of successfully transferred blocks for a multiple block transfer. | R  |

### 32.6.11 MSC Interrupt Mask Register (MSC\_IMASK)

| MSC_IMASK |          |                 |     |           |    |    |    |          |    |    |    |     |     |     |     | 0x24   |                 |                |               |             |              |               |              |               |      |               |               |          |             |          |                |   |
|-----------|----------|-----------------|-----|-----------|----|----|----|----------|----|----|----|-----|-----|-----|-----|--------|-----------------|----------------|---------------|-------------|--------------|---------------|--------------|---------------|------|---------------|---------------|----------|-------------|----------|----------------|---|
| Bit       | 31       | 30              | 29  | 28        | 27 | 26 | 25 | 24       | 23 | 22 | 21 | 20  | 19  | 18  | 17  | 16     | 15              | 14             | 13            | 12          | 11           | 10            | 9            | 8             | 7    | 6             | 5             | 4        | 3           | 2        | 1              | 0 |
|           | Reserved | AUTO_CMD23_DONE | SVS | PIN_LEVEL |    |    |    | Reserved |    |    |    | BCE | BDE | BAE | BAR | DMAEND | AUTO_CMD12_DONE | DATA_FIFO_FULL | DATA_FIFO_EMP | CRC_RES_ERR | CRC_READ_ERR | CRC_WRITE_ERR | TIME_OUT_RES | TIME_OUT_READ | SDIO | TXFIFO_WR_REQ | RXFIFO_RD_REQ | Reserved | END_CMD_RES | PRG_DONE | DATA_TRAN_DONE |   |
| RST       | 1        | 1               | 1   | 1         | 1  | 1  | 1  | 1        | 1  | 1  | 1  | 1   | 1   | 1   | 1   | 1      | 1               | 1              | 1             | 1           | 1            | 1             | 1            | 1             | 1    | 1             | 1             | 1        | 1           | 1        | 1              |   |

| Bits | Name            | Description                                                       | RW |
|------|-----------------|-------------------------------------------------------------------|----|
| 31   | Reserved        | Writing has no effect.                                            | R  |
| 30   | AUTO_CMD23_DONE | Mask the interrupt AUTO_CMD23_DONE.<br>0: Not masked<br>1: Masked | RW |
| 29   | SVS             | Mask the interrupt of SVS<br>0: Not masked<br>1: Masked           | RW |

|       |                 |                                                                               |    |
|-------|-----------------|-------------------------------------------------------------------------------|----|
| 28:24 | PIN_LEVEL       | Mask the interrupt of PIN_LEVEL separately<br>0: Not masked<br>1: Masked      | RW |
| 23:21 | Reserved        | Writing has no effect, read as zero.                                          | R  |
| 20    | BCE             | Boot CRC error.<br>0: Not masked<br>1: Masked                                 | RW |
| 19    | BDE             | Boot data end.<br>0: Not masked<br>1: Masked                                  | RW |
| 18    | BAE             | Mask the interrupt of BAE.<br>0: Not masked.<br>1: Masked.                    | RW |
| 17    | BAR             | Mask the interrupt of BAR.<br>0: Not masked.<br>1: Masked.                    | RW |
| 16    | DMAEND          | Mask the interrupt of DMA end.<br>0: Not masked.<br>1: Masked.                | RW |
| 15    | AUTO_CMD12_DONE | Mask the interrupt AUTO_CMD12_DONE.<br>0: Not masked<br>1: Masked             | RW |
| 14    | DATA_FIFO_FULL  | 0: Not masked<br>1: Masked                                                    | RW |
| 13    | DATA_FIFO_EMP   | 0: Not masked<br>1: Masked                                                    | RW |
| 12    | CRC_RES_ERR     | 0: Not masked<br>1: Masked                                                    | RW |
| 11    | CRC_READ_ERR    | 0: Not masked<br>1: Masked                                                    | RW |
| 10    | CRC_WRITE_ERR   | 0: Not masked<br>1: Masked                                                    | RW |
| 9     | TIME_OUT_RES    | 0: Not masked<br>1: Masked                                                    | RW |
| 8     | TIME_OUT_READ   | 0: Not masked<br>1: Masked                                                    | RW |
| 7     | SDIO            | Mask the interrupt from the SD I/O card.<br>0: Not masked<br>1: Masked        | WR |
| 6     | TXFIFO_WR_REQ   | Mask the Transmit FIFO write request interrupt.<br>0: Not masked<br>1: Masked | WR |

|     |                |                                                                             |    |
|-----|----------------|-----------------------------------------------------------------------------|----|
| 5   | RXFIFO_RD_REQ  | Mask the Receive FIFO read request interrupt.<br>0: Not masked<br>1: Masked | WR |
| 4:3 | Reserved       | Writing has no effect, read as zero.                                        | R  |
| 2   | END_CMD_RES    | Mask the End command response interrupt.<br>0: Not masked<br>1: Masked      | WR |
| 1   | PRG_DONE       | Mask the Programming done interrupt.<br>0: Not masked<br>1: Masked          | WR |
| 0   | DATA_TRAN_DONE | Mask the Data transfer done interrupt.<br>0: Not masked<br>1: Masked        | WR |

### 32.6.12 MSC Interrupt Flag Register (MSC\_IFLG)

The MSC\_IFLG register shows the currently requested interrupt. The FIFO request interrupts, TXFIFO\_WR\_REQ, and RXFIFO\_RD\_REQ are masked off with the DMA function is used. The software is responsible for monitoring these bit in program I/O mode.

| MSC_IFLG |          |                 |     |           |    |    |    |          |     |     |     |     |        |                 |                | 0x28          |             |              |               |              |               |      |               |               |          |             |          |                |   |   |   |   |
|----------|----------|-----------------|-----|-----------|----|----|----|----------|-----|-----|-----|-----|--------|-----------------|----------------|---------------|-------------|--------------|---------------|--------------|---------------|------|---------------|---------------|----------|-------------|----------|----------------|---|---|---|---|
| Bit      | 31       | 30              | 29  | 28        | 27 | 26 | 25 | 24       | 23  | 22  | 21  | 20  | 19     | 18              | 17             | 16            | 15          | 14           | 13            | 12           | 11            | 10   | 9             | 8             | 7        | 6           | 5        | 4              | 3 | 2 | 1 | 0 |
|          | Reserved | AUTO_CMD23_DONE | SVS | PIN_LEVEL |    |    |    | Reserved | BCE | BDE | BAE | BAR | DMAEND | AUTO_CMD12_DONE | DATA_FIFO_FULL | DATA_FIFO_EMP | CRC_RES_ERR | CRC_READ_ERR | CRC_WRITE_ERR | TIME_OUT_RES | TIME_OUT_READ | SDIO | TXFIFO_WR_REQ | RXFIFO_RD_REQ | Reserved | END_CMD_RES | PRG_DONE | DATA_TRAN_DONE |   |   |   |   |
| RST      | 0        | 0               | 0   | 0         | 0  | 0  | 0  | 0        | 0   | 0   | 0   | 0   | 0      | 0               | 0              | 0             | 0           | 0            | 1             | 0            | 0             | 0    | 0             | 0             | 0        | 0           | 0        | 0              | 0 | 0 | 0 | 0 |

| Bits  | Name            | Description                                                                                                       | RW |
|-------|-----------------|-------------------------------------------------------------------------------------------------------------------|----|
| 31    | Reserved        | Writing has no effect, read as zero.                                                                              | R  |
| 30    | AUTO_CMD23_DONE | Indicates AUTO_CMD23_DONE interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected          | RW |
| 29    | SVS             | Indicates SVS interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected<br>Write 1 to clear. | RW |
| 28:24 | PIN_LEVEL       | Indicates PIN_LEVEL interrupt.<br>0: the interrupt is not detected                                                | RW |

|       |                 |                                                                                                                           |    |
|-------|-----------------|---------------------------------------------------------------------------------------------------------------------------|----|
|       |                 | 1: the interrupt is detected<br>Write 1 to clear.                                                                         |    |
| 23:21 | Reserved        | Writing has no effect, read as zero.                                                                                      | R  |
| 20    | BCE             | Boot CRC error.<br>0: the interrupt is not detected<br>1: the interrupt is detected<br>Write 1 to clear.                  | RW |
| 19    | BDE             | Boot data end.<br>0: the interrupt is not detected<br>1: the interrupt is detected<br>Write 1 to clear.                   | RW |
| 18    | BAE             | Indicates the BAE interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected<br>Write 1 to clear.     | RW |
| 17    | BAR             | Indicates the BAR interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected<br>Write 1 to clear.     | RW |
| 16    | DMAEND          | Indicates the DMA end interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected<br>Write 1 to clear. | RW |
| 15    | AUTO_CMD12_DONE | Indicates AUTO_CMD12_DONE interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected                  | RW |
| 14    | DATA_FIFO_FULL  | Indicate data FIFO is full interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected                 | R  |
| 13    | DATA_FIFO_EMP   | Indicate data FIFO is empty interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected                | R  |
| 12    | CRC_RES_ERR     | Indicate response CRC error interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected                | RW |
| 11    | CRC_READ_ERR    | Indicate CRC read error interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected                    | RW |
| 10    | CRC_WRITE_ERR   | Indicate CRC write error interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected                   | RW |
| 9     | TIME_OUT_RES    | Indicate response time out interrupt.<br>0: the interrupt is not detected                                                 | RW |



|     |                |                                                                                                                                                                                                                                                               |    |
|-----|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|     |                | 1: the interrupt is detected                                                                                                                                                                                                                                  |    |
| 8   | TIME_OUT_READ  | Indicate read time out interrupt.<br>0: the interrupt is not detected<br>1: the interrupt is detected                                                                                                                                                         | RW |
| 7   | SDIO           | Indicates whether the interrupt from SDIO is detected.<br>0: The interrupt from SDIO is not detected<br>1: The interrupt from SDIO is detected                                                                                                                | R  |
| 6   | TXFIFO_WR_REQ  | Transmit FIFO write request. Set if data FIFO becomes half empty. (the number of words is < 8)<br>0: No Request for data Write to MSC_TXFIFO<br>1: Request for data write to MSC_TXFIFO                                                                       | R  |
| 5   | RXFIFO_RD_REQ  | Receive FIFO read request. Set if data FIFO becomes half full (the number of words is >= 8) or the entries in data FIFO are the last read data.<br>0: No Request for data read from MSC_RXFIFO<br>1: Request for data read from MSC_RXFIFO                    | R  |
| 4:3 | Reserved       | Writing has no effect, read as zero.                                                                                                                                                                                                                          | R  |
| 2   | END_CMD_RES    | Indicates whether the command/response sequence has been finished.<br>0: The command/response sequence has not been finished<br>1: The command/response sequence has been finished<br>Write 1 to clear.                                                       | WR |
| 1   | PRG_DONE       | Indicates whether card has finished programming.<br>0: Card has not finished programming and is busy<br>1: Card has finished programming and is no longer busy<br>Write 1 to clear.                                                                           | WR |
| 0   | DATA_TRAN_DONE | Indicates whether data transfer is done.<br>Note that for stream read/write, only when CMD12 (STOP_TRANS) has been sent, is this bit set.<br>0: Data transfer is not complete<br>1: Data transfer has completed or an error has occurred<br>Write 1 to clear. | WR |

### 32.6.13 MSC Command Index Register (MSC\_CMD)

|         |  |  |  |  |  |  |  |  |  |  |  |  |  |      |          |  |           |   |   |   |   |   |   |   |
|---------|--|--|--|--|--|--|--|--|--|--|--|--|--|------|----------|--|-----------|---|---|---|---|---|---|---|
| MSC_CMD |  |  |  |  |  |  |  |  |  |  |  |  |  | 0x2C |          |  |           |   |   |   |   |   |   |   |
| Bit     |  |  |  |  |  |  |  |  |  |  |  |  |  |      |          |  | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|         |  |  |  |  |  |  |  |  |  |  |  |  |  |      | Reserved |  | CMD_INDEX |   |   |   |   |   |   |   |
| RST     |  |  |  |  |  |  |  |  |  |  |  |  |  |      |          |  | 0         | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name      | Description                                 | RW |
|------|-----------|---------------------------------------------|----|
| 7:6  | Reserved  | Writing has no effect, read as zero.        | R  |
| 5:0  | CMD_INDEX | Specifies the command index to be executed. | RW |

### 32.6.14 MSC Command Argument Register (MSC\_ARG)

|                |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|----------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| <b>MSC_ARG</b> |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x30</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>     | 31  | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16          | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|                | ARG |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>     | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name | Description                                     | RW |
|------|------|-------------------------------------------------|----|
| 31:0 | ARG  | Specifies the argument for the current command. | RW |

### 32.6.15 MSC Response FIFO Register (MSC\_RES)

The read-only MMC/SD Response FIFO register (RES\_FIFO) holds the response sent back to the MMC/SD controller after every command. The size of this FIFO is 8 x 16-bit. The RES\_FIFO does not contain the 7-bit CRC for the response. The Status for CRC checking and response time-out status is in the status register MSC\_STAT.

The first half-word read from the response FIFO is the most significant half-word of the received response.

|                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |      |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>MSC_RES</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x34</b> |      |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>Bit</b>     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |      | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             | DATA |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| <b>RST</b>     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |      | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? |   |

| Bits | Name | Description                                                                 | RW |
|------|------|-----------------------------------------------------------------------------|----|
| 15:0 | DATA | Contains the response to the command that is sent by the MMC/SD controller. | R  |

### 32.6.16 MSC Receive FIFO Port Register (MSC\_RXFIFO)

The MSC\_RXFIFO is used to read the data from a card. It is read-only to the software, and is read on 32-bit boundary. The size of this FIFO is 128 x 32-bit.

|                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|-------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| <b>MSC_RXFIFO</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <b>0x38</b> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|-------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|

|     |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
|-----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| Bit | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |
|     | DATA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |
| RST | ?    | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description            | RW |
|------|------|------------------------|----|
| 31:0 | DATA | One word of read data. | R  |

### 32.6.17 MSC Transmit FIFO Port Register (MSC\_TXFIFO)

The MSC\_TXFIFO is used to write the data to a card. It is write-only to the software, and is written on 32-bit boundary. The size of this FIFO is 128 x 32-bit.

|                   |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------------------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| <b>MSC_TXFIFO</b> |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x3C</b> |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit               | 31   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16          | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|                   | DATA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RST               | ?    | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?  | ?           | ?  | ?  | ?  | ?  | ?  | ?  | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

| Bits | Name | Description             | RW |
|------|------|-------------------------|----|
| 31:0 | DATA | One word of write data. | W  |

### 32.6.18 MSC Low Power Mode Register (MSC\_LPM)

The MSC\_LPM is used to control whether MSC controller enters Low-Power Mode.

|                |         |         |          |    |    |    |    |    |    |    |    |    |    |    |    |             |    |    |    |    |    |    |   |   |   |   |     |   |   |   |   |   |
|----------------|---------|---------|----------|----|----|----|----|----|----|----|----|----|----|----|----|-------------|----|----|----|----|----|----|---|---|---|---|-----|---|---|---|---|---|
| <b>MSC_LPM</b> |         |         |          |    |    |    |    |    |    |    |    |    |    |    |    | <b>0x40</b> |    |    |    |    |    |    |   |   |   |   |     |   |   |   |   |   |
| Bit            | 31      | 30      | 29       | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16          | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5   | 4 | 3 | 2 | 1 | 0 |
|                | DRV_SEL | SMP_SEL | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |             |    |    |    |    |    |    |   |   |   |   | LPM |   |   |   |   |   |
| RST            |         |         | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 |     | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name    | Description                                                                                                                                                                                                                                                            | RW |
|-------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:30 | DRV_SEL | Drive clock selection<br>0: CMD and data are driven by MSC_CLK falling edge<br>1: CMD and data are driven by clock rising edge which is 1ns delayed by MSC_CLK<br>2: CMD and data are driven by clock rising edge which is 1/4 phase delayed by MSC_CLK<br>3: Reserved | RW |

|      |          |                                                                                                                                                                                                                                                                                                                                                                                                   |    |
|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|      |          | When this field is not zero, MSC_CLKRT.CLK_RATE must be set to 0.                                                                                                                                                                                                                                                                                                                                 |    |
| 29   | SMP_SEL  | Sample clock selection<br>0: CMD and data are sampled by MSC_CLK rising edge<br>1: CMD and data are sampled by clock rising edge which is 1/4 or 1/2 phase delayed by MSC_CLK.<br>When this field is 1, MSC_CLKRT.CLK_RATE must be set to 0.                                                                                                                                                      |    |
| 28:1 | Reserved | Writing has no effect, read as zero.                                                                                                                                                                                                                                                                                                                                                              | R  |
| 0    | LPM      | Low Power Mode Enable<br>0: Disable low power mode<br>1: Enable low power mode<br>Clock will stop when card in idle (should be normally set to only MMC and SD cards. For SDIO cards, if interrupts must be detected, clock should not be stopped)<br>When software sets the bit, MSC clock can auto be stopped.<br><b>NOTE:</b> when set the bit, the start_clock and stop clock can be not use. | RW |

### 32.6.19 MSC DMA Control Register (MSC\_DMAC)

The MSC\_DMAC register is used to control the DMA transfer.

|            |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |       |         |      |        |       |   |   |   |   |   |   |   |   |   |   |             |
|------------|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|-------|---------|------|--------|-------|---|---|---|---|---|---|---|---|---|---|-------------|
|            | <b>MSC_DMAC</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |       |         |      |        |       |   |   |   |   |   |   |   |   |   |   | <b>0x44</b> |
| <b>Bit</b> | 31              | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14    | 13      | 12   | 11     | 10    | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |             |
|            | Reserved        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | MODE_SEL | AOFST | ALIGNEN | INCR | DMASEL | DMAEN |   |   |   |   |   |   |   |   |   |   |             |
| <b>RST</b> | 0               | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0     | 0       | 0    | 0      | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |             |

| Bits | Name     | Description                                                                                                                                                                                                                                                                                                                                     | RW |
|------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:8 | Reserved | Writing has no effect.<br>Read as zero.                                                                                                                                                                                                                                                                                                         | R  |
| 7    | MODE_SEL | Transfer mode selection.<br>0: Do not specify the transfer length. This is the standard transfer.<br>1: Specify the transfer length. If this mode is selected, the transfer performance can be improved.                                                                                                                                        |    |
| 6:5  | AOFST    | Address Offset.<br>This field is effective only when ALIGNEN is 1b.<br>00b: The lowest 2 bit of the data transfer start address is 00b<br>01b: The lowest 2 bit of the data transfer start address is 01b<br>10b: The lowest 2 bit of the data transfer start address is 10b<br>11b: The lowest 2 bit of the data transfer start address is 11b | RW |
| 4    | ALIGNEN  | Align Enable                                                                                                                                                                                                                                                                                                                                    | RW |

|     |        |                                                                                                                                                                                                                    |    |
|-----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|     |        | 0: Only word boundary data transfer is supported<br>1: Byte boundary data transfer is supported                                                                                                                    |    |
| 3:2 | INCR   | Burst type selection.<br>The field only effects to special DMA in MSC controller.<br>2'b00: The burst type of DMA is INCR16.<br>2'b01: The burst type of DMA is INCR32.<br>2'b10: The burst type of DMA is INCR64. | RW |
| 1   | DMASEL | One of supported DMA modes is selected.<br>0: Special DMA in MSC controller is used<br>1: Common DMA in DMAC controller is used                                                                                    | RW |
| 0   | DMAEN  | DMA Function Enable<br>0: Disable DMA function<br>1: Enable DMA function                                                                                                                                           | RW |

### 32.6.20 MSC DMA Descriptor Address Register (MSC\_DMANDA)

|            |                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |             |   |
|------------|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-------------|---|
|            | <b>MSC_DMANDA</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | <b>0x48</b> |   |
| <b>Bit</b> | 31                | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |             |   |
|            | NDA               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |             |   |
| <b>RST</b> | 0                 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0           | 0 |

| Bits | Name | Description                                                                                                                              | RW |
|------|------|------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | NDA  | SDMA Next Descriptor Physical Address<br>The SDMA will change the register when a descriptor is read in.<br>It should be 4-word aligned. | RW |

### 32.6.21 MSC DMA Data Address Register (MSC\_DMADA)

|            |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |             |   |
|------------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-------------|---|
|            | <b>MSC_DMADA</b> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | <b>0x4c</b> |   |
| <b>Bit</b> | 31               | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |             |   |
|            | DA               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |             |   |
| <b>RST</b> | 0                | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0           | 0 |

| Bits | Name | Description                                                                                                                                                                  | RW |
|------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | DA   | SDMA Data Physical Address<br>The data address of the current descriptor will be copied to this field.<br>The SDMA will increment it during the data transfer automatically. | R  |

### 32.6.22 MSC DMA Data Length Register (MSC\_DMALEN)

|            |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSC_DMALEN |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x50 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Bit        | 31  | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16   | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|            | LEN |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| RST        | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits | Name | Description                                                                                                                                                                                              | RW |
|------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:0 | LEN  | SDMA Data Length to be Transferred<br>Its unit is byte.<br>The data length of the current descriptor will be copied to this field.<br>The SDMA will decrement it during the data transfer automatically. | R  |

### 32.6.23 MSC DMA Command Register (MSC\_DMACMD)

|            |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |        |          |   |   |   |   |      |      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----------|----|----|----|--------|----------|---|---|---|---|------|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSC_DMACMD |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0x54 |          |    |    |    |        |          |   |   |   |   |      |      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Bit        | 31  | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16   | 15       | 14 | 13 | 12 | 11     | 10       | 9 | 8 | 7 | 6 | 5    | 4    | 3 | 2 | 1 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|            | IDI |    |    |    |    |    |    |    | ID |    |    |    |    |    |    |      | Reserved |    |    |    | OFFSET | ALIGN_EN |   |   |   |   | ENDI | LINK |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| RST        | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0        | 0  | 0  | 0  | 0      | 0        | 0 | 0 | 0 | 0 | 0    | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bits  | Name     | Description                                                                                                                                                                                            | RW |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:24 | IDI      | Identification of the Interrupt DMA.<br>This field will copy the ID when a DMA transfer interrupt occurs.<br>If the interrupt of DMA is disabled, the field will not change.                           | R  |
| 23:16 | ID       | Identification of Current DMA Transfer.                                                                                                                                                                | R  |
| 15:11 | Reserved | It has no use in the current version.                                                                                                                                                                  | R  |
| 10:9  | OFFSET   | Address Offset.<br>This field is effective when ALIGN_EN is set to 1b.<br>00: data address[1:0] is 00b<br>01: data address[1:0] is 01b<br>10: data address[1:0] is 10b<br>11: data address[1:0] is 11b | R  |
| 8     | ALIGN_EN | Address Align Enable.<br>This bit should be set to 1b if the data address is not word-aligned.                                                                                                         | R  |
| 7:2   | Reserved | It has no use in the current version.                                                                                                                                                                  | R  |
| 1     | ENDI     | Interrupt Enable for Current DMA Transfer End<br>0: Disable interrupt                                                                                                                                  | R  |

|   |      |                                                                                                                                                                                                                                                                                                                                                                 |   |
|---|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|   |      | 1: Enable interrupt                                                                                                                                                                                                                                                                                                                                             |   |
| 0 | LINK | Control the end of DMA Descriptor.<br>0: DMA will go to idle state after the current transmission is finished. The MSC_STAT.DMAEND will be set to 1b when the current read/write is finished. If the interrupt mask bit MSC_IMASK.DMAEND is 0b, the interrupt flag MSC_IFLG.DMAEND will be set to 1b also.<br>1: DMA will continue to fetch another descriptor. | R |

### 32.6.24 MSC RTFIFO Data Counter Register(MSC\_RTCNT)

|     |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
|-----|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------|
|     | MSC_RTCNT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | 0x5c |
| Bit | 31        | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |      |
|     | RTCNT     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
| RST | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    |

| Bits | Name  | Description                                                            | RW |
|------|-------|------------------------------------------------------------------------|----|
| 31:0 | RTCNT | This field indicates how many data in word units are stored in RTFIFO. | R  |

### 32.7 MMC/SD Functional Description

All communication between system and cards is controlled by the MSC. The MSC sends commands of two type: broadcast and addressed (point-to-point) commands.

Broadcast commands are intended for all cards, command like “Go\_Idle\_State”, “Send\_Op\_Cond”, “All\_send\_CID” and “Set\_relative\_Addr” are using way of broadcasting. During Broadcast mode, all cards are in open-drain mode, to avoid bus contention.

After Broadcast commands “Set\_relative\_Addr” issue, cards are enter standby mode, and Addressed command will be used from now on, in this mode, CMD/DAT will return to push-pull mode, to have maximum driving for maximum operation frequency.

The MMC and the SD are similar product. Besides the 4x bandwidth and the built-in encryption, they are being programmed similarly.

The MMC/SD controller (MSC) is the interface between the software and the MMC/SD bus. It is responsible for the timing and protocol between the software and the MMC/SD bus. It consists of control and status registers, a 16-bit response FIFO that is 8 entries deep, and one 32-bit receive/transmit data FIFOs that are 16 entries deep. The registers and FIFOs are accessible by the software.

MSC also enable minimal data latency by buffering data and generating and checking CRCs.

### 32.7.1 MSC Reset

The MMC/SD controller (MSC) can be reset by a hardware reset or software reset. All registers and FIFO controls are set to their default values after any reset.

### 32.7.2 MSC Card Reset

The command Go\_Idle\_State, CMD0 is the software reset command for MMC and SD Memory Card, and sets each card into Idle State regardless of the current card state; while in SDIO card, CMD52 is used to write IO reset in CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

### 32.7.3 Voltage Validation

All cards shall be able to establish communication with the host using any operation voltage in the maximal allowed voltage range specified in this standard. However, the support minimum and maximum values for Vdd are defined in Operation Conditions register (OCR) and many not cover the whole range. Cards that store the CID and CSD data in the payload memory would be able to communicate these information only under data transfer Vdd conditions. That means if host and card have non compatible Vdd ranges, the card will not be able to complete the identification cycle, nor to send CSD data.

Therefore, a special command Send\_Op\_cont (CMD1 for MMC), SD\_Send\_Op\_Cont (CMD41 for SD Memory) and IO\_Send\_Op\_Cont (CMD5 for SDIO) are designed to provide a mechanism to identify and reject cards which do not match the Vdd range desired by the host. This is accomplished by the host sending the required Vdd voltage window as the operand of this command. Cards which can not perform data transfer in the specified range must discard themselves from further bus operations and go into Inactive State. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive State. This query should be used if the host is able to select a common voltage range or if a notification to the application of non usable cards in the stack is desired.

### 32.7.4 Card Registry

Card registry on MCC and SD card are different.

For SD card, Identification process start at clock rate Fod, while CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated the host will request the cards to send their valid operation conditions. The response to ACMD41 is the operation condition register of the card.



The same command shall be send to all of the new cards in the system. Incompatible cards are sent into Inactive State. The host then issue the command All\_Send\_CID (CMD2) to each card and get its unique card identification (CID) number. Card that is unidentified, that is, which is in Ready State, send its CID number as the response. After the CID was sent by the card it goes into Identification State. Thereafter, the host issues Send\_Relative\_Addr (CMD3) asks the card to publish a new relative card address (RCA), which is shorter that CID and which will be used to address the card in the future data transfer mode. Once the RCA is received the card state changes to the Stand-by State. At this point, if the host wants that the card will have another RCA number, it may ask the card to publish a new number by sending another Send\_Relative\_Addr command to the card. The last published RCA is the actual RCA of the card. The host repeats the identification process, that is, the cycles with CMD2 and CMD3 for each card in the system.

In MMC, the host starts the card identification process in open-drain mode with the identification clock rate Fod. The open drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is active the host will request the cards to send their valid operation conditions (CMD1). The response to CMD1 is the 'wired or' operation on the condition restrictions of all cards in the system. Incompatible cards are sent into Inactive State. The host then issues the broadcast command All\_Send\_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards, that is, those which are in Ready State, simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bitstream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods stop sending their CID immediately and must wait for the next identification cycle. Since CID is unique for each card, only one card can be successfully send its full CID to the host. This card then goes into Identification State. Thereafter, the host issues Set\_Relative\_Addr (CMD3) to assign to this card a relative card address (RCA). Once the RCA is received the card state changes to the Stand-by State, and the card does not react to further identification cycles, and its output switches from open-drain to push-pull. The host repeat the process, which is CM2 and CMD3, until the host receive time-out condition to recognize completion of the identification process.

## 32.7.5 Card Access

### 32.7.5.1 Block Access, Block Write and Block Read

During block write (CMD24-27) one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. A card supporting block write shall always be able to accept a block of data defined by WRITE\_BL\_LEN. If the CRC fails, the card shall indicate the failure on the DAT line; the transferred data will be discarded and not written, and all further transmitted blocks (in multiple block write mode) will be ignored.

Programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents. Some cards may require long and

unpredictable times to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT line low if its write buffer is full and unable to accept new data from a new WRITE\_BLOCK command. The host may poll the status of the card with a SEND\_STATUS command (CMD13) at any time, and the card will respond with its status. The status bit READY\_FOR\_DATA indicates whether the card can accept new data or whether the write process is still in progress). The host may deselect the card by issuing CMD7 (to select a different card) which will displace the card into the Disconnect State and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling DAT to low if programming is still in progress and the write buffer is unavailable.

Block read is similar to stream read, except the basic unit of data transfer is a block whose maximizes is defined in the CSD (READ\_BL\_LEN). If READ\_BL\_PARTIAL is set, smaller blocks whose starting and ending address are entirely contained within one physical block (as defined by READ\_BL\_LEN) may also be transmitted. Unlike stream read, a CRC is appended to the end of each block ensuring data transfer integrity. CMD17 (READ\_SINGLE\_BLOCK) initiates a block read and after completing the transfer, the card returns to the Transfer state. CMD18 (READ\_MULTIPLE\_BLOCK) starts a transfer of several consecutive blocks. Blocks will be continuously transferred until a stop command is issued. If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card shall detect a block misalignment at the beginning of the first mis-aligned block, set the ADDRESS\_ERROR error bit in the status register, abort transmission and wait in the Data State for a stop command.

### 32.7.5.2 Stream Access, Stream Write and Stream Read (MMC Only)

Stream write (CMD20) starts the data transfer from the host to the card beginning from the starting address until the host issues a stop command. Since the amount of data to be transferred is not determined in advance, CRC can not be used. If the end of the memory range is reached while sending data and no stop command has been sent by the host, all further transferred data is discarded.

There is a stream oriented data transfer controlled by READ\_DAT\_UNTIL\_STOP (CMD11). This command instructs the card to send its payload, starting at a specified address, until the host sends a STOP\_TRANSMISSION command (CMD12). The stop command has execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command. If the end of the memory range is reached while sending data and no stop command has been sent yet by the host, the contents of the further transferred payload is undefined.

### 32.7.5.3 Erase, Group Erase and Sector Erase (MMC Only)

It is desirable to erase many sectors simultaneously in order to enhance the data throughput. Identification of these sectors is accomplished with the TAG\_\* commands. Either an arbitrary set of sectors within a single erase group, or an arbitrary selection of erase groups may be erase at one time, but not both together. That is, the unit of measure for determining an erase is either a sector or an

erase group. If a set of sectors must be erased, all selected sectors must lie within the same erase group. To facilitate selection, a first command with the starting address is followed by a second command with the final address, and all sectors (or groups) within this range will be selected for erase.

#### **32.7.5.4 Wide Bus Selection/Deselection**

Wide Bus (4 bit bus width) operation mode may be selected / deselected using ACMD6. The default bus width after power up or GO\_IDLE (CMD0) is 1 bit bus width. ACMD6 command is valid in 'trans state' only. That means the bus width may be changed only after a card was selected (CMD7).

### **32.7.6 Protection Management**

Three write protect methods are supported in the host for Cards, Card internal write protect (Card's responsibility), Mechanical write protect switch (Host responsibility only) and Password protection card lock operation.

#### **32.7.6.1 Card Internal Write Protection**

Card data may be protected against either erase or write. The entire card may be permanently write protected by the manufacturer or content provider by setting the permanent or temporary write protect bits in the CSD. For cards which support write protection of groups of sectors by setting the WP\_GRP\_SIZE sectors as specified in the CSD), and the write protection may be changed by the application. The SET\_WRITE\_PROT command sets the write protection of the addressed write-protect group, and the CLR\_WRITE\_PROT command clears the write protection of the addressed write-protect group.

The SEND\_WRITE\_PROT command is similar to a single block read command. The card shall send a data block containing 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits. The address field in the write protect commands is a group address in byte units. The card will ignore all LSB's below the group size.

#### **32.7.6.2 Mechanical write protect switch**

A mechanical sliding tablet on the side of the card will be used by the user to indicate that a given card is write protected or not. If the sliding tablet is positioned in such a way that the window is open that means the card is write protected. If the window is close the card is not write protected.

A proper, matched, switch on the socket side will indicated to the host that the card is write protected or not. It is the responsibility of the host to protect the card. The position of the write protect switch is un-known to the internal circuitry of the card.

### 32.7.6.3 Password Protect

The password protection feature enables the host to lock a card while providing a password, which later will be used for unlocking the card. The password and its size is kept in an 128-bit PWD and 8-bit PWD\_LEN registers, respectively. These registers are non-volatile so that a power cycle will not erase them.

Locked cards respond to (and execute) all commands in the basic command class (class 0) and “lock card” command class. Thus the host is allowed to reset, initialize, select, query for status, etc., but not to access data on the card. If the password was previously set (the value of PWD\_LEN is not 0) will be locked automatically after power on. Similar to the existing CSD and CID register write commands the lock/unlock command is available in “trans\_state” only. This means that it does not include an address argument and the card must be selected before using it. The card lock/unlock command has the structure and bus transaction type of a regular single block write command. The transferred data block includes all the required information of the command (password setting mode, PWD itself, card lock/unlock etc.). The following table describes the structure of the command data block.

**Table 32-4 Command Data Block Structure**

| Byte #   | Bit 7         | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2       | Bit 1   | Bit 0   |
|----------|---------------|-------|-------|-------|-------|-------------|---------|---------|
| 0        | Rsv           | Rsv   | Rsv   | Rsv   | ERASE | LOCK_UNLOCK | CLR_PWD | SET_PWD |
| 1        | PWDS_LEN      |       |       |       |       |             |         |         |
| 2        | Password Data |       |       |       |       |             |         |         |
| ...      |               |       |       |       |       |             |         |         |
| PWDS_LEN |               |       |       |       |       |             |         |         |
| + 1      |               |       |       |       |       |             |         |         |

**ERASE** – 1 Defines Forced Erase Operation (all other bits shall be 0) and only the command byte is sent.

**LOCK/UNLOCK** – 1=Locks the card. 0=Unlock the card (note that it is valid to set this bit together with SET\_PWD but it is not allowed to set it together with CLR\_PWD).

**CLR\_PWD** – 1=Clears PWD.

**SET\_PWD** – 1=Set new password to PWD.

**PWD\_LEN** – Defines the following password length (in bytes).

**PWD** – The password (new or currently used depending on the command).

The data block size shall be defined by the host before it send the card lock/unlock command. This will allow different password sizes.

The following paragraphs define the various lock/unlock command sequences:

Lock command sequences:

- 1 Setting the Password.
  - a Select a card (CMD7), if not previously selected already.
  - b Define the block length (CMD16), given by the 8bit card lock/unlock mode, the 8 bits

password size (in bytes), and the number of bytes of the new password. In case that a password replacement is done, then the block size shall consider that both passwords, the old and the new one, are sent with the command.

- c Send Card Lock/Unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode (SET\_PWD), the length (PWD\_LEN) and the password itself. In case that a password replacement is done, then the length value (PWD\_LEN) shall include both passwords, the old and the new one, and the PWD field shall include the old password (currently used) followed by the new password.
- d In case that the sent old password is not correct (not equal in size and content) then LOCK\_UNLOCK\_FAILED error bit will be set in the status register and the old password does not change. In case that PWD matches the sent old password then the given new password and its size will be saved in the PWD and PWD\_LEN fields, respectively.

**NOTE:**

the password length register (PWD\_LEN) indicates if a password is currently set. When it equals 0 there is no password set. If the value of PWD\_LEN is not equal to zero the card will lock itself after power up. It is possible to lock the card immediately in the current power session by setting the LOCK/UNLOCK bit (while setting the password) or sending additional command for card lock.

- 2 Reset the password.
  - a Select a card (CMD7), if not previously selected already.
  - b Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
  - c Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode CLR\_PWD, the length (PWD\_LEN) and the password (PWD) itself (LOCK/UNLOCK bit is don't care). If the PWD and PWD\_LEN is set to 0. If the password is not correct then the LOCK\_UNLOCK\_FAILED error bit will be set in the status register.
- 3 Locking a card.
  - a Select a card (CMD7), if not previously selected already.
  - b Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of currently used password.
  - c Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode LOCK, the length (PWD\_LEN) and the password (PWD) itself.

If the PWD content equals to the sent password then the card will be locked and the card-locked status bit will be set in the status register. If the password is not correct then LOCK\_UNLOCK\_FAILED error bit will be set in the status register.

**NOTE:**

it is possible to set the password and to lock the card in the same sequence. In such case the host shall perform all the required steps for setting the password (as described above) including the bit LOCK set while the new password command is sent. If the password was previously set (PWD\_LEN is not 0), then the card will be locked automatically after power on reset. An attempt to lock a locked card or to lock a card that does not have a password will fail and the LOCK\_UNLOCK\_FAILED error bit will be set in the status register.

Unlock command sequences:

- 1 Unlocking the card.
  - a Select a card (CMD7), if not previously selected already.
  - b Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
  - c Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode UNLOCK, the length (PWD\_LEN) and the password (PWD) itself.

If the PWD content equals to the sent password then the card will be unlocked and the card-locked status bit will be cleared in the status register. If the password is not correct then the LOCK\_UNLOCK\_FAILED error bit will be set in the status register.

**NOTE:**

the unlocking is done only for the current power session. As long as the PWD is not cleared the card will be locked automatically on the next power up. The only way to unlock the card is by clearing the password. An attempt to unlock an unlocked card will fail and LOCK\_UNLOCK\_FAILED error bit will be set in the status register.

- 2 Forcing Erase.

In case that the user forgot the password (the PWD content) it is possible to erase all the card data content along with the PWD content. This operation is called Forced Erase.

- a Select a card (CMD7), if not previously selected already.
- b Define the block length (CMD16) to 1 byte (8bit card lock/unlock command). Send the card lock/unlock command with the appropriate data block of one byte on the data line including 16-bit CRC. The data block shall indicate the mode ERASE (the ERASE bit shall be the only bit set).

If the ERASE bit is not the only bit in the data field then the LOCK\_UNLOCK\_FAILED error bit will be set in the status register and the erase request is rejected. If the command was accepted then ALL THE CARD CONTENT WILL BE ERASED including the PWD and PWD\_LEN register content and the locked card will get unlocked.

An attempt to force erase on an unlocked card will fail and LOCK\_UNLOCK\_FAILED error bit will be set in the status register.

### 32.7.7 Card Status

The response format R1 contains a 32-bit field named card status. This field is intended to transmit the card's status information (which may be stored in a local status register) to the host. If not specified otherwise, the status entries are always related to the previous issued command.

Table below defines the different entries of the status. The type and clear condition fields in the table are abbreviate as follows:

Type:

- E: Error bit.
- S: Status bit..
- R: Detected and set for the actual command response.
- X: Detected and set during command execution. The host must poll the card by issuing the status command in order to read these bits.

Clear Condition:

- A: According to the card current state.
- B: Always related to the previous command. Reception of a valid command will clear it (with a delay of one command).
- C: Clear by read.

**Table 32-5 Card Status Description**

| Bits | Identifier      | Type  | Description                                                                                                                                          | Clear Condition |
|------|-----------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 31   | OUT_OF_RANGE    | E R   | The command's argument was out of the allowed range for this card.<br>0: No Error<br>1: Error                                                        | C               |
| 30   | ADDRESS_ERROR   | E R X | A misaligned address which did not match the block length was used in the command.<br>0: No Error<br>1: Error                                        | C               |
| 29   | BLOCK_LEN_ERROR | E R   | The transferred block length is not allowed for this, or the number of transferred bytes does not match the block length.<br>0: No Error<br>1: Error | C               |



|    |                    |       |                                                                                                                                                                    |   |
|----|--------------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 28 | ERASE_SEQ_ERROR    | E R   | An error in the sequence of erase commands occurred.<br>0: No Error<br>1: Error                                                                                    | C |
| 27 | ERASE_PARAM        | E X   | An invalid selection of sectors or groups for erase occurred.<br>0: No Error<br>1: Error                                                                           | C |
| 26 | WP_VIOLATION       | E R X | Attempt to program a write protected block.<br>0: No Protected<br>1: Protected                                                                                     | C |
| 25 | CARD_IS_LOCKED     | S X   | When set, signals that the card is locked by the host.<br>0: Card unlocked<br>1: Card locked                                                                       | A |
| 24 | LOCK_UNLOCK_FAILED | E R X | Set when a sequence or password error has been detected in lock/unlock card command or if there was an attempt to access a locked card.<br>0: No Error<br>1: Error | C |
| 23 | COM_CRC_ERROR      | E R   | The CRC check of the previous command failed.<br>0: No Error<br>1: Error                                                                                           | B |
| 22 | ILLEGAL_COMMAND    | E R   | Command not legal for the card state.<br>0: No Error<br>1: Error                                                                                                   | B |
| 21 | CARD_ECC_FAILED    | E X   | Card internal ECC was applied but failed to correct the data.<br>0: normal<br>1: failure                                                                           | C |
| 20 | CC_ERROR           | E R X | Internal card controller error.<br>0: No Error<br>1: Error                                                                                                         | C |
| 19 | ERROR              | E R X | A general or an unknown error occurred during the operation.<br>0: No Error<br>1: Error                                                                            | C |
| 18 | UNDERRUN           | E X   | The card could not sustain data                                                                                                                                    | C |



|      |                   |       |                                                                                                                                                                                                                                                                                                                                                                   |   |
|------|-------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|      |                   |       | transfer in stream read mode.<br>0: No Error<br>1: Error                                                                                                                                                                                                                                                                                                          |   |
| 17   | OVERRUN           | E X   | The card could not sustain data programming in stream write mode.<br>0: No Error<br>1: Error                                                                                                                                                                                                                                                                      | C |
| 16   | CID/CSD_OVERWRITE | E R X | Can be either one of the following errors.<br>0: No Error<br>1: Error                                                                                                                                                                                                                                                                                             | C |
| 15   | WP_ERASE_SKIP     | S X   | Only partial address space was erased due to existing write protected blocks.<br>0: No Protected<br>1 : Protected                                                                                                                                                                                                                                                 | C |
| 14   | CARD_ECC_DISABLED | S X   | The command has been executed without using the internal ECC.<br>0: enabled<br>1: disabled                                                                                                                                                                                                                                                                        | A |
| 13   | ERASE_RESET       | S R   | An erase sequence was cleared before executing because an out of erase sequence command was received.<br>0: normal<br>1: set                                                                                                                                                                                                                                      | C |
| 12:9 | CURRENT_STATE     | S X   | The state of the card when receiving the command. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are interpreted as binary coded number between 0 and 15.<br>0: idle<br>1: ready<br>2: ident<br>3: stby<br>4: tran<br>5: data<br>6: rcv<br>7: prg<br>8 : dis<br>(9 – 15) : rsv | B |

|     |                |     |                                                                                                                    |   |
|-----|----------------|-----|--------------------------------------------------------------------------------------------------------------------|---|
| 8   | READY_FOR_DATA | S X | Corresponds to buffer empty signaling on the bus.<br>0: No Ready<br>1: Ready                                       | A |
| 7:6 | Reserved       | -   | -                                                                                                                  | - |
| 5   | APP_CMD        | S R | The card will expect ACMD, or indication that the command has been interpreted as ACMD.<br>0: Disable<br>1: Enable | C |
| 4:0 | Reserved       | -   | -                                                                                                                  | - |

### 32.7.8 SD Status

The SD status contains status bits that are related to the SD card proprietary features and may be used for future application specific usage. The size of the SD status is one data block of 512bit. The content of this register is transmitted to the Host over the DAT bus along with 16-bit CRC. The SD status is sent to the host over the DAT bus if ACMD13 is sent (CMD55 followed with CMD13). ACMD13 can be sent to a card only in 'tran\_state' (card is selected). SD status structure is described in below.

The same abbreviation for *type* and *clear condition* were used as for the Card Status above.

**Table 32-6 SD Status Structure**

| Bits    | Identifier                 | Type | Description                                                                                                                                                 | Clear Condition |
|---------|----------------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 511:510 | DAT_BUS_WIDTH              | S R  | Shows the currently defined data bus width that was defined by SET_BUS_WIDTH command.<br>00: 1 (default)<br>01: Reserved<br>10: 4 bit width<br>11: Reserved | A               |
| 509     | SECURED_MODE               | S R  | Card is in Secured Mode of operation.<br>0: Not in the Mode<br>10: In the mode                                                                              | A               |
| 508:496 | Reserved.                  |      |                                                                                                                                                             |                 |
| 495:480 | SD_CARD_TYPE               | S R  | All 0, is SD Memory cards.                                                                                                                                  | A               |
| 479:448 | SIZE_OF_PROTECTED_AREA     | S R  | Size of protected area.                                                                                                                                     | A               |
| 447:312 | Reserved.                  |      |                                                                                                                                                             |                 |
| 311:0   | Reserved for manufacturer. |      |                                                                                                                                                             |                 |

### 32.7.9 SDIO

I/O access differs from memory in that the registers can be written and read individually and directly without a FAT file structure or the concept of blocks (although block access is supported). These registers allow access to the IO data, control of the IO function, and report on status or transfer I/O data to and from the host.

Each SDIO card may have from 1 to 7 functions plus one memory function built into it. A function is a self contained I/O device. I/O functions may be identical or completely different from each other. All I/O functions are organized as a collection of registers, and there is a maximum of 131,072 registers possible for each I/O function.

#### 32.7.9.1 SDIO Interrupts

In order to allow the SDIO card to interrupt the host, an interrupt function is added to a pin on the SD interface. Pin number 8 which is used as DAT[1] when operating in the 4 bit SD mode is used to signal the card's interrupt to the host. The use of interrupt is optional for each card or function within a card. The SDIO interrupt is "level sensitive", that is, the interrupt line must be held active (low) until it is either recognized and acted upon by the host or de-asserted due to the end of the Interrupt Period. Once the host has serviced the interrupt, it is cleared via an IO write to the appropriate bit in the CCCR. The interrupt output of all SDIO cards is active low.

As Pin 8 of the card is shared between the IRQ and DAT[1] use in the 4 bit SD mode, and interrupt shall only be sent by the card and recognized by the host during a specific time. The time that a low on Pin 8 will be recognized as an interrupt is defined as the Interrupt Period.

The host here will only sample the level of Pin 8 (DAT[1]/IRQ) into the interrupt detector during the Interrupt Period. At all other times, the host will ignore the level on Pin 8. Note that the Interrupt Period is applicable for both memory and IO operations. The definition of the Interrupt Period is different for operations with single block and multiple block data transfer.

#### 32.7.9.2 SDIO Suspend/Resume

Within a multi-function SDIO or a Combo (Mix IO and Memory) card, there are multiple devices (I/O and memory) that must share access to the SD bus. In order to allow the sharing of access to the host among multiple devices, SDIO and combo cards can implement the optional concept of suspend/resume. In a card supports suspend/resume, the host may temporarily halt a data transfer operation to one function or memory (suspend) in order to free the bus for a higher priority transfer to a different function or memory. Once this higher-priority transfer is complete, the original transfer is re-started where it left off (resume). The host controller here is supported by all IO functions except zero, and the memory of a combo card, and can suspend multiple transactions and resume them in any order desired. IO function zero does not support suspend/resume.

The procedure used to perform the Suspend/Resume operation on the SD bus is:

- The host determines which function currently used the DAT[] line(s).
- The host requests the lower priority or slower transaction to suspend.
- The host checks for the transaction suspension to complete.
- The host begins the higher priority transaction.
- The host waits for the completion of the higher priority transaction.
- The host restores the suspended transaction.

### 32.7.9.3 SDIO Read Wait

The optional Read Wait (RW) operation is defined only for the SD 1-bit and 4-bit modes. The read wait operation allows a host to signal a card that it is doing a read multiple (CMD53) operation to temporarily stall the data transfer while allowing the host to send commands to any function within the SDIO device. To determine if a card supports the Read Wait protocol, the host must test capability bits in CCCR. The timing for Read Wait is base on the Interrupt Period.

### 32.7.10 Clock Control

The software should guarantee that the card identification process starts in open-drain mode with the clock rate fod (0 ~ 400khz). In addition, the software should also make the card into interrupt mode with fod (only for MMC). The commands that require fod are CMD0, CMD1, CMD2, CMD3, CMD5, CMD40 and ACMD41. In data transfer mode, the MSC controller can operate card with clock rate fpp (0 ~ 25Mhz).

### 32.7.11 Application Specified Command Handling

The MultiMediaCard/SD system is designed to provide a standard interface for a variety applications types. In this environment it is anticipate that there will be a need for specific customers/applications features. To enable a common way of implementing these features, two types of generic commands are defined in the standard: Application Specific Command, ACMD, and General Command, GEN\_CMD.

GEN\_CMD, this command, when received by the card, will cause the card to interpret the following command as an application specific command, ACMD. The ACMD has the same structure as of regular MultiMediaCard standard commands and it may have the same CMD number. The card will recognize it as ACMD by the fact that it appears after APP\_CMD.

The only effect of the APP\_CMD is that if the command index of the, immediately, following command has an ACMD overloading, the none standard version will used. If, as an example, a card has a definition for ACMD13 but not for ACMD7 then, if received immediately after APP\_CMD command, Command 13 will be interpreted as the non standard ACMD13 but, command 7 as the standard CMD7.

In order to use one of the manufacturer specific ACMD's the host will:

- 1 Send APP\_CMD. The response will have the APP\_CMD bit (new status bit) set signaling to the host that ACMD is now expected.
- 2 Send the required ACMD. The response will have the APP\_CMD bit set, indicating that the accepted command was interpreted as ACMD. If a non-ACMD is sent then it will be respected by the card as normal MultiMediaCard command and the APP\_CMD bit in the Card Status stays clear.

If a non valid command is sent (neither ACMD nor CMD) then it will be handled as a standard MultiMediaCard illegal command error.

The bus transaction of the GEN\_CMD is the same as the single block read or write commands (CMD24 or CMD17). The difference is that the argument denotes the direction of the data transfer (rather than the address) and the data block is not a memory payload data but has a vendor specific format and meaning.

The card shall be selected ('tran\_state') before sending CMD56. The data block size is the BLOCK\_LEN that was defined with CMD16. The response to CMD56 will be R1b (card status + busy indication).

## 32.8 MMC/SD Controller Operation

### 32.8.1 Data FIFOs

The controller FIFOs for the response tokens, received data, and transmitted data are MSC\_RES, MSC\_RXFIFO, and MSC\_TXFIFO, respectively. These FIFOs are accessible by the software and are described in the following paragraphs.

#### 32.8.1.1 Response FIFO (MSC\_RES)

The response FIFO, MSC\_RES, contains the response received from an MMC/SD card after a command is sent from the controller. MSC\_RES is a read-only, 16-bit, and 8-entry deep FIFO.

The FIFO will hold all possible response lengths. Responses that are only one byte long are located on the LSB of the 16-bit entry in the FIFO. The first half-word read from the response FIFO is the most significant half-word of the received response. For example, if the response format is R1, then the response read from RES\_FIFO is bit [47:32], bit[31:16], bit[15:0] and in the third half-word only the low 8-bit is effective response [15:8] and the high 8-bit is ignored. If the response format is R2, then the response read from MSC\_RES is bit [135:8] and needs reading 8 times.

The FIFO does not contain the response CRC. The status of the CRC check is in the status register, MSC\_STAT.

### 32.8.1.2 Receive/Transmit Data FIFO (MSC\_RXFIFO/MSC\_TXFIFO)

The receive data FIFO and transmit data FIFO share one 16-entry x 32-bit FIFO, because at one time data are only received or are only transmitted. If it is used to receive data, it is called MSC\_RXFIFO and read-only. If it is used to transmit data, it is called MSC\_TXFIFO and write-only.

Data FIFO and its controls are cleared to a starting state after a system reset or at the beginning of the operations which include data transfer. (MSC\_CMDAT[DATA\_EN] == 1)

If at any time MSC\_RXFIFO becomes full and the data transmission is not complete, the controller turns the MSC\_CLK off to prevent any overflows. When the clock is off, data transmission from the card stops until the clock is turned back on. After MSC\_RXFIFO is not full, the controller turns the clock on to continue data transmission. The full status of the FIFO is registered in the MSC\_STAT [DATA\_FIFO\_FULL] bit.

If at any time MSC\_TXFIFO becomes empty and the data transmission is not complete, the controller turns the MSC\_CLK off to prevent any underrun. When the clock is off, data transmission to the card stops until the clock is turned back on. When MSC\_TXFIFO is no longer empty, the controller automatically restarts the clock. The empty status of the FIFO is registered in the MSC\_STAT [DATA\_FIFO\_EMPTY] bit.

The FIFO is readable on word (32-bit) boundaries. The max read/written number is 16 words. The controller can correctly process big-endian and little-endian data.

Because at the beginning of the operation which include data transfer (MSC\_CMDAT [DATA\_EN] == 1), Data FIFO and its controls are cleared, software should guarantee data in FIFO have been read/written before beginning a new command.

## 32.8.2 DMA and Program I/O

Software may communicate to the MSC controller via the DMA or program I/O. The SDMA and CDMA are supported at the same time. SDMA is the special DMA in MSC controller, while CDMA is the common DMA in DMAC controller.

### 32.8.2.1 Structure of SDMA Descriptor

SDMA is a descriptor DMA. And a descriptor can be linked to another one by the NDA until the LINK in the 4<sup>th</sup> word is set to 1b.

A SDMA descriptor includes 4 words.

Word 1: The physical address of the next SDMA descriptor.

| Bits | Name | Description                                                       |
|------|------|-------------------------------------------------------------------|
| 31:0 | NDA  | Next descriptor physical address.<br>It should be 4-word aligned. |

Word 2: The system memory address where the data will be read from or stored to.

| Bits | Name | Description   |
|------|------|---------------|
| 31:0 | DA   | Data Address. |

Word 3: The length of data to be transferred.

| Bits | Name | Description   |
|------|------|---------------|
| 31:0 | LEN  | Unit is byte. |

Word 4: The command for the current SDMA.

| Bits  | Name     | Description                                                                                                                                                                                                                                                                                                                                        |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:24 | Reserved | It has no use in the current version.                                                                                                                                                                                                                                                                                                              |
| 23:16 | ID       | Identification of Current DMA Transfer.                                                                                                                                                                                                                                                                                                            |
| 15:2  | Reserved | It has no use in the current version.                                                                                                                                                                                                                                                                                                              |
| 1     | ENDI     | Interrupt Enable for Current DMA Transfer End<br>0: Disable interrupt<br>1: Enable interrupt                                                                                                                                                                                                                                                       |
| 0     | LINK     | Control the end of DMA Descriptor.<br>0: DMA will continue to fetch another descriptor.<br>1: DMA will go to idle state after the current transfer.<br>The MSC_STAT.DMAEND will be set to 1b when the current read/write is finished. If the interrupt mask bit MSC_IMASK.DMAEND is 0b, the interrupt flag MSC_IFLG.DMAEND will be set to 1b also. |

### 32.8.2.2 Operation of SDMA

Step 1: Prepare the descriptor in system memory

Step 2: If the address is not word boundary, it is suggested to configure MSC\_DMAC.ALIGNEN and MSC\_DMAC.AOFST

Step 3: Write the descriptor address to MSC\_DMANDA

Step 4: Configure MSC controller to read or write operation, e.g, set MSC\_NOB, MSC\_BLEN and MSC\_CMDAT.WRITE\_READ etc.

Step 5: If DMAEND interrupt is wanted, clear MSC\_IMASK.DMAEND, otherwise, set this bit

Step 6: Select SDMA by setting MSC\_DMAC.DMASEL to 0b

Step 7: Start DMA transfer by setting MSC\_DMAC.DMAEN

Step 8: Waiting the DMAEND interrupt (if interrupt is used) or status (if interrupt is not used)

Step 9: Disable the SDMA by configuring MSC\_DMAC.DMAEN to 0b

### 32.8.2.3 Operation of CDMA

To access MSC\_RXFIFO/MSC\_TXFIFO with the DMA, the software must program the DMA to read or write the FIFO with source port width 32-bit, destination port width 32-bit, transfer data size 32-byte, transfer mode single. For example, to write 64 bytes of data to the MSC\_TXFIFO, the software must program the DMA as follows:

```

DMA_DCTRn = 2 // Write 2 32-bytes (64 bytes)
DMA_DCCRn[SWDH] = 0 // source port width is 32-bit
DMA_DCCRn[DWDH] = 0 // destination port width is 32-bit
DMA_DCCRn[DS] = 4 // transfer data size is 32-byte
DMA_DCCRn[TM] = 4 // transfer mode is single
DMA_DCCRn[RDIL] = 0 // request detection interval length is 0

```

The number of 32-bytes should be calculated from the number of transferred bytes as follows:

The number of words = (The number of bytes + 31) / 32

If the number of transferred bytes is not the multiple of 4, the controller can correctly process endian.

The DMA trigger level is 8 words, that is to say, the DMA read trigger is when data words in MSC\_RXFIFO is  $\geq 8$  and the DMA write trigger is when data words in MSC\_TXFIFO is  $< 8$ . Software can also configure DMA registers based on requirements, but the above 32-byte transfer data size is most efficient.

### 32.8.2.4 Operation of Program I/O

With program I/O, the software waits for the MSC\_IFLG [RXFIFO\_RD\_REQ] or MSC\_IFLG [TXFIFO\_WR\_REQ] interrupts before reading or writing the respective FIFO.

#### NOTES:

- 1 The MSC\_CMDAT [DMA\_EN] bit must be set to a 1 to enable communication with the DMA and it must be set to a 0 to enable program I/O.
- 2 DMA can be enabled only after MSC\_CMDAT is written, because MSC\_CMDAT [DATA\_EN] is used to reset TX/RXFIFO.

### 32.8.3 Start and Stop clock

The software stops the clock as follows:

- 1 Write MSC\_CTRL with 0x01 to stop the MMC/SD bus clock.
- 2 Wait until MSC\_STAT[CLK\_EN] becomes zero.

To start the clock the software writes MSC\_CTRL with 0x02.

### 32.8.4 Software Reset

Reset includes the MSC reset and the card reset.



The MSC reset is through MSC\_CTRL [RESET] bit.

The card reset is to make the card into idle state. CMD0 (GO\_IDLE\_STATE) sets the MMC and SD memory cards into idle state. CMD52 (IO\_RW\_DIRECT, with argument 0x88000C08) reset the SD I/O card. The MMC/SD card are initialized with a default relative card address (RCA = 0x0001 for MMC and RCA = 0x0000 for SD) and with a default driver stage register setting (lowest speed, highest driving current capability).

The following registers must be set before the clock is started:

- Step 1. Stop the clock.
- Step 2. Set MSC\_CTRL register to 0x08 to reset MSC.
- Step 3. Wait while MSC\_STAT [IS\_RESETTING] is 1.
- Step 4. Set MSC\_CMD with CMD0.
- Step 5. Update the MSC\_CMDAT register as follows:
  - a Write 0x0000 to MSC\_CMDAT [RESPONSE\_FORMAT].
  - b Clear the MSC\_CMDAT [DATA\_EN] bit.
  - c Clear the MSC\_CMDAT [BUSY] bit.
  - d Clear the MSC\_CMDAT [INIT] bit.
- Step 6. Start the clock.
- Step 7. Start the operation. (write MSC\_CTRL with 0x04)
- Step 8. Wait for the END\_CMD\_RES interrupt.
- Step 9. Set MSC\_CMD with CMD52.
- Step 10. Set MSC\_ARG with 0x88000C08.
- Step 11. Update the MSC\_CMDAT register as follows:
  - a Write 0x005 to MSC\_CMDAT [RESPONSE\_FORMAT].
  - b Clear the MSC\_CMDAT [DATA\_EN] bit.
  - c Clear the MSC\_CMDAT [BUSY] bit.
  - d Clear the MSC\_CMDAT [INIT] bit.
- Step 12. Start the operation.
- Step 13. Wait for the END\_CMD\_RES interrupt.

### 32.8.5 Voltage Validation and Card Registry

At most 10 MMC and 1 SD (either SDMEM or SDIO) can be inserted MMC/SD bus at the same time, and their voltage validation and card registry steps are different, so the software should be programmed as follows:

- Step 1. Check whether SDIO card is inserted.
- Step 2. Check whether SDMEM card is inserted.
- Step 3. Check whether MMC cards are inserted.

### 32.8.5.1 Check SDIO

The commands are sent as follows:

- Step 1. (Optional) Send CMD52 (IO\_RW\_DIRECT) with argument 0x88000C08 to reset SDIO card.
- Step 2. Send CMD5 (IO\_SEND\_OP\_CMD) to validate voltage.
- Step 3. If the response is correct and the number of IO functions > 0, then continue, else go to check SDMEM.
- Step 4. If C-bit in the response is ready (the initialization has finished), go to 6.
- Step 5. Send CMD5 (IO\_SEND\_OP\_CMD) to validate voltage, then go to 4.
- Step 6. If memory-present-bit in the response is true, then it is a combo card (SDIO + Memory), else it is only a SDIO card.
- Step 7. If it is a combo card, go to check SDMEM to initialize the memory part.
- Step 8. Send CMD3 (SET\_RELATIVE\_ADDR) to let the card publish a RCA. The RCA is returned from the response.
- Step 9. If do not accept the new RCA, go to 8, else record the new RCA.
- Step 10. Go to check MMC, because we can assure that there is no SDMEM card.

### 32.8.5.2 Check SDMEM

If there is no SDIO card or there is a combo card, continue to check SDMEM.

The commands are sent as follows:

- Step 1. (Optional) Send CMD0 (GO\_IDLE\_STATE) to reset MMC and SDMEM card. This command has no response.
- Step 2. Send CMD55. Here the default RCA 0x0000 is used for CMD55.
- Step 3. If the response is correct (CMD55 has response), then continue, else go to check MMC.
- Step 4. Send ACMD41 (SD\_SEND\_OP\_CMD) to validate voltage (the general OCR value is 0x00FF8000).
- Step 5. If the initialization has finished, go to 7. (The response is the OCR register and it includes a status information bit (bit [31]). This status bit is set if the card power up procedure has been finished. As long as the card is busy, the corresponding bit[31] is set to LOW.)
- Step 6. Send CMD55 and ACMD41 to validate voltage, and then go to 5.
- Step 7. Send CMD2 (ALL\_SEND\_CID) to get the card CID.
- Step 8. Send CMD3 (SET\_RELATIVE\_ADDR) to let card publish a RCA. The RCA is returned from the response.
- Step 9. If do not accept the new RCA, go to 8, else record the new RCA.
- Step 10. Go to check MMC.

### 32.8.5.3 Check MMC

Because there may be several MMC card, so some steps (5 ~ 8) should be repeated several times.

The commands are sent as follows:

- Step 1. Send CMD1 (SEND\_OP\_CMD) to validate voltage (the general OCR value is 0x00FF88000).
- Step 2. If the response is correct, then continue, else goto 9.
- Step 3. If the initialization has finished, go to 5. (The response is the OCR register and it includes a status information bit (bit [31]). This status bit is set if the card power up procedure has been finished. As long as the card is busy, the corresponding bit[31] is set to LOW.)
- Step 4. Send CMD1 (SEND\_OP\_CMD) to validate voltage, and then go to 3.
- Step 5. Send CMD2 (ALL\_SEND\_CID) to get the card CID.
- Step 6. If the response timeout occurs, goto 9.
- Step 7. Send CMD3 (SET\_RELATIVE\_ADDR) to assign the card a RCA.
- Step 8. If there are other MMC cards, then go to 5.
- Step 9. Finish.

### 32.8.6 Single Data Block Write

In a single block write command, the following registers must be set before the operation is started:

- Step 1. Set MSC\_NOB register to 0x0001.
- Step 2. Set MSC\_BLKLEN to the number of bytes per block.
- Step 3. Update the MSC\_CMDAT register as follows:
  - a Write 0x001 to MSC\_CMDAT [RESPONSE\_FORMAT].
  - b Write 0x2 to MSC\_CMDAT [BUS\_WIDTH] if the card is SD, else clear it.
  - c Set the MSC\_CMDAT [DATA\_EN] bit.
  - d Set the MSC\_CMDAT [WRITE\_READ] bit.
  - e Clear the MSC\_CMDAT [STREAM\_BLOCK] bit.
  - f Clear the MSC\_CMDAT [BUSY] bit.
  - g Clear the MSC\_CMDAT [INIT] bit.
- Step 4. Start the operation.
- Step 5. Write MSC\_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

- Step 1. Wait for the MSC\_IFLG [END\_CMD\_RES] interrupt.
- Step 2. Wait for the MSC\_IFLG [DATA\_TRAN\_DONE] interrupt.

At the same time write data to the MSC\_TXFIFO and continue until all of the data have been written to the FIFO.
- Step 3. Wait for MSC\_IFLG [PROG\_DONE] interrupt. This interrupt indicates that the card has finished programming. Certainly software may start another command sequence on a different card.
- Step 4. Read the MSC\_STAT register to verify the status of the transaction (i.e. CRC error status).

To address a different card, the software sends a select command to that card by sending a basic no data command and response transaction. To address the same card, the software must wait for MSC\_IFLG [PROG\_DONE] interrupt. This ensures that the card is not in the busy state.

In addition, CMD26 (PROGRAM\_CID), CMD27 (PROGRAM\_CSD), CMD42 (LOCK/UNLOCK), CMD56 (GEN\_CMD: write) and CMD53 (single\_block\_write) operations are similar to single block write.

### 32.8.7 Single Block Read

In a single block read command, the following registers must be set before the operation is started:

- Step 1. Set MSC\_NOB register to 0x0001.
- Step 2. Set MSC\_BLKLEN register to the number of bytes per block.
- Step 3. Update the following bits in the MSC\_CMDAT register:
  - a Write 0x001 to MSC\_CMDAT [RESPONSE\_FORMAT].
  - b Write 0x2 to MSC\_CMDAT [BUS\_WIDTH] if the card is SD, else clear it.
  - c Set the MSC\_CMDAT [DATA\_EN] bit.
  - d Clear the MSC\_CMDAT [WRITE\_READ] bit.
  - e Clear the MSC\_CMDAT [STREAM\_BLOCK] bit.
  - f Clear the MSC\_CMDAT [BUSY] bit.
  - g Clear the MSC\_CMDAT [INIT] bit.
- Step 4. Start the operation.
- Step 5. Write MSC\_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

- Step 1. Wait for the MSC\_IFLG [END\_CMD\_RES] interrupt.
- Step 2. Wait for the MSC\_IFLG [DATA\_TRAN\_DONE] interrupt.

At the same time read data from the MSC\_RXFIFO as data becomes available in the FIFO, and continue reading until all data is read from the FIFO.
- Step 3. Read the MSC\_STAT register to verify the status of the transaction (i.e. CRC error status).

In addition, CMD30 (SEND\_WRITE\_PROT), ACMD13 (SD\_STATUS), CMD56 (GEN\_CMD-read), ACMD51 (SEND\_SCR) and CMD53 (single\_block\_read) are similar to single block read.

### 32.8.8 Multiple Block Write

The multiple block write mode is similar to the single block write mode, except that multiple blocks of data are transferred. Each block is the same length. All the registers are set as they are for the single block write, except that the MSC\_NOB register is set to the number of blocks to be written.

The multiple block write mode also requires a stop transmission command, CMD12, after the data is transferred to the card. After the MSC\_IFLG [DATA\_TRAN\_DONE] interrupt occurs, the software must program the controller register to send a stop data transmission command.

If multiple block write with pre-defined block count (refer to MMC spec v-3.3) is used, CMD12 should not be sent.

For SDIO card, CMD53 (multiple\_block\_write) is also similar, but when IO abort (CMD52) is sent, MSC\_CMDAT [IO\_ABORT] should be 1.

**Table 32-7 How to stop multiple block write**

| Operation                       | Stop condition                                     | Software processing                                                                                                                                                                                                                |
|---------------------------------|----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Open-ended or SDIO infinite     | After write MSC_NOB blocks                         | <ol style="list-style-type: none"> <li>1 Wait for DATA_TRAN_DONE interrupt.</li> <li>2 Send CMD12 or CMD52. (IO abort)</li> <li>3 Wait for END_CMD_RES and PRG_DONE interrupt.</li> </ol>                                          |
| Open-ended or SDIO infinite     | Stop writing in advance (not write MSC_NOB blocks) | <ol style="list-style-type: none"> <li>1 Set MSC_CTRL [EXIT_MULTIPLE].</li> <li>2 Wait for DATA_TRAN_DONE interrupt.</li> <li>3 Send CMD12 or CMD52. (IO abort)</li> <li>4 Wait for END_CMD_RES and PRG_DONE interrupt.</li> </ol> |
| Predefined block or SDIO finite | After writing MSC_NOB blocks                       | <ol style="list-style-type: none"> <li>1 Wait for DATA_TRAN_DONE interrupt.</li> </ol>                                                                                                                                             |
| Predefined block or SDIO finite | Stop writing in advance (not write MSC_NOB blocks) | <ol style="list-style-type: none"> <li>1 Set MSC_CTRL [EXIT_MULTIPLE].</li> <li>2 Wait for DATA_TRAN_DONE interrupt.</li> <li>3 Send CMD12 or CMD52. (IO abort)</li> <li>4 Wait for END_CMD_RES and PRG_DONE interrupt.</li> </ol> |

### 32.8.9 Multiple Block Read

The multiple blocks read mode is similar to the single block read mode, except that multiple blocks of data are transferred. Each block is the same length. All the registers are set as they are for the single block read, except that the MSC\_NOB register is set to the number of blocks to be read.

The multiple blocks read mode requires a stop transmission command, CMD12, after the data from the card is received. After the MSC\_IFLG [DATA\_TRAN\_DONE] interrupt has occurred, the software must program the controller registers to send a stop data transmission command.

If multiple block read with pre-defined block count (refer to MMC spec v-3.3) is used, CMD12 should not be sent.

For SDIO card, CMD53 (multiple\_block\_read) is also similar, but when IO abort (CMD52) is sent, MSC\_CMDAT [IO\_ABORT] should be 1.

**Table 32-8 How to stop multiple block read**

| Operation                       | Stop condition                                     | Software processing                                                                                                                               |
|---------------------------------|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Open-ended or SDIO infinite     | After reading MSC_NOB blocks                       | 1 Wait for DATA_TRAN_DONE interrupt.<br>2 Send CMD12 or CMD52. (IO abort)<br>3 Wait for END_CMD_RES interrupt.                                    |
| Open-ended or SDIO infinite     | Stop reading in advance (not write MSC_NOB blocks) | 1 Set MSC_CTRL [EXIT_MULTIPLE].<br>2 Wait for DATA_TRAN_DONE interrupt.<br>3 Send CMD12 or CMD52. (IO abort)<br>4 Wait for END_CMD_RES interrupt. |
| Predefined block or SDIO finite | After reading MSC_NOB blocks                       | 1 Wait for DATA_TRAN_DONE interrupt.                                                                                                              |
| Predefined block or SDIO finite | Stop reading in advance (not write MSC_NOB blocks) | 1 Set MSC_CTRL [EXIT_MULTIPLE].<br>2 Wait for DATA_TRAN_DONE interrupt.<br>3 Send CMD12 or CMD52. (IO abort)<br>4 Wait for END_CMD_RES interrupt. |

### 32.8.10 Stream Write (MMC)

In a stream write command, the following registers must be set before the operation is started:

- 1 Update MSC\_CMDAT register as follows:
  - a Write 0x001 to the MSC\_CMDAT [RESPONSE\_FORMAT].
  - b Clear the MSC\_CMDAT [BUS\_WIDTH] because only MMC support stream write.
  - c Set the MSC\_CMDAT [DATA\_EN] bit.
  - d Set the MSC\_CMDAT [WRITE\_READ] bit.
  - e Set the MSC\_CMDAT [STREAM\_BLOCK] bit.
  - f Clear the MSC\_CMDAT [BUSY] bit.
  - g Clear the MSC\_CMDAT [INIT] bit.
- 2 Start the operation.
- 3 Write MSC\_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

- 1 Wait for the MSC\_IFLG [END\_CMD\_RES] interrupt.
- 2 Write data to the MSC\_TXFIFO and continue until all of the data is written to the Data FIFO.
- 3 Stop clock. Wait until MSC\_STAT[CLK\_EN] becomes 0. The clock must be stopped.
- 4 Set the command registers for a stop transaction command (CMD12) and other registers.
- 5 Start the clock and start the operation.
- 6 Wait for the MSC\_IFLG [END\_CMD\_ERS] interrupt.
- 7 Wait for the MSC\_IFLG [DATA\_TRAN\_DONE] interrupt.
- 8 Wait for the MSC\_IFLG [PRG\_DONE] interrupt. This interrupt indicates that the card has finished programming. Certainly software may start another command sequence on a different card.
- 9 Read the MSC\_STAT register to verify the status of the transaction.

To address a different card, the software must send a select command to that card by sending a basic no data command and response transaction. To address the same card, the software must wait for MSC\_IFLG [PRG\_DONE] interrupt. This ensures that the card is not in the busy state.

If partial blocks are allowed (if CSD parameter WRITE\_BL\_PARTIAL is set) the data stream can start and stop at any address within the card address space, otherwise it shall start and stop only at block boundaries. If WRITE\_BL\_PARTIAL is not set, 16 more stuff bytes need to be written after the useful written data, otherwise only write the useful written data.

### 32.8.11 Stream Read (MMC)

In a stream read command, the following registers must be set before the operation is turned on:

- 1 Update the MSC\_CMDAT register as follows:
  - a Write 0x01 to the MSC\_CMDAT [RESPONSE\_FORMAT].
  - b Clear the MSC\_CMDAT [BUS\_WIDTH] because only MMC support stream read.
  - c Clear the MSC\_CMDAT [WRITE\_READ] bit.
  - d Set the MSC\_CMDAT [STREAM\_BLOCK] bit.
  - e Clear the MSC\_CMDAT [BUSY] bit.
  - f Clear the MSC\_CMDAT [INIT] bit.
- 2 Start the operation.
- 3 Write MSC\_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

- 1 Wait for the MSC\_IFLG [END\_CMD\_RES] interrupt.
- 2 Read data from the MSC\_RXFIFO and continue until all of the expected data has been read from the FIFO.
- 3 Write MSC\_CTRL [EXIT\_TRANSER] with 1. If MSC\_STAT[DATA\_FIFO\_FULL] is 1, then read MSC\_RXFIFO to make it not full. Because if data FIFO is full, MSC\_CLK is stopped. Here, the data FIFO contains useless data.
- 4 Set the command registers for a stop transaction command (CMD12) and send it. There is no need to stop the clock.
- 5 Wait for the MSC\_IFLG [END\_CMD\_RES].
- 6 Wait for the MSC\_IFLG [DATA\_TRAN\_DONE] interrupt.
- 7 Read the MSC\_STAT register to verify the status of the transaction.

### 32.8.12 Erase, Select/Deselect and Stop

For CMD7 (SELECT/DESELECT\_CARD), CMD12 (STOP\_TRANSMISSION) and CMD38 (ERASE), the following registers must be set before the operation is started:

- 1 Update the MSC\_CMDAT register as follows:
  - a Write 0x01 to the MSC\_CMDAT [RESPONSE\_FORMAT].
  - b Clear the MSC\_CMDAT [DATA\_EN] bit.



- c Clear the MSC\_CMDAT [WRITE\_READ] bit.
  - d Clear the MSC\_CMDAT [STREAM\_BLOCK] bit.
  - e Set the MSC\_CMDAT [BUSY] bit.
  - f Clear the MSC\_CMDAT [INIT] bit.
- 2 Start the operation.
  - 3 Write MSC\_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

- 1 Wait for the MSC\_IFLG [END\_CMD\_RES] interrupt.
- 2 Wait for the MSC\_IFLG [PRG\_DONE] interrupt. If CMD12 is sent to terminate data read operation, then there is no need to wait for MSC\_IFLG [PRG\_DONE] interrupt. This interrupt indicates that the card has finished programming. Certainly software may start another command sequence on a different card.

### 32.8.13 SDIO Suspend/Resume

The actual suspend/resume steps are as follows:

- 1 During data transfer, send CMD52 to require suspend. BR and RAW flag should be 1.
- 2 If BS flag in the response is 0, then suspend has been accepted and goto 4.
- 3 Send CMD52 to query card status. R flag should be 1. Go to 2.
- 4 Write MSC\_CTRL [EXIT\_TRANSFER] with 1.
- 5 Wait for the MSC\_IFLG [DATA\_TRAN\_DONE] interrupt.
- 6 Read MSC\_NOB, MSC\_SNOB and etc, save them into variables.
- 7 Set registers for high priority transfer and start it.
- 8 Wait until high priority transfer is finished.
- 9 Restore registers from variables, but MSC\_NOB should be (MSC\_NOB – MSC\_SNOB).
- 10 Send CMD52 to require resume. FSx should be resumed function number.

### 32.8.14 SDIO ReadWait

The actual ReadWait steps are as follows:

- 1 During multiple block read, read MSC\_SNOB. If MSC\_SNOB is nearby or equal to MSC\_NOB, no need to use ReadWait.
- 2 Write MSC\_CTRL [START\_READWAIT] with 1.
- 3 Wait until MSC\_STAT [IS\_READWAIT] becomes 1.
- 4 Send CMD52 to query card status.
- 5 Write MSC\_CTRL [STOP\_READWAIT] with 1.

### 32.8.15 Operation and Interrupt

The software can use polling-status method to operate the MMC/SD card, but this is not the proposed method, because its performance is very low. The proposed method is to use interrupt. Generally there are fixed necessary steps to finish each command. The steps are as follows:

- 1 (Optional) Stop clock. Poll CLK\_EN.



- 2 Fill the registers (MSC\_CMD, MSC\_CMDAT, MSC\_ARG, MSC\_CLKRT, and etc).
- 3 (Optional) Start clock.
- 4 Start the operation. Wait for the MSC\_IFLG [END\_CMD\_RES] interrupt.
- 5 Wait for the MSC\_IFLG [DATA\_TRAN\_DONE] interrupt.
- 6 Send STOP\_TRANS (CMD12) or I/O abort (CMD52). Wait for the MSC\_IFLG [END\_CMD\_ERS] interrupt.
- 7 Wait for the MSC\_IFLG [DATA\_TRAN\_DONE] interrupt.
- 8 Wait for the MSC\_IFLG [PRG\_DONE] interrupt.

**Table 32-9 The mapping between Commands and Steps**

| Index | Abbreviation         | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Comments         |
|-------|----------------------|---|---|---|---|---|---|---|---|------------------|
| CMD0  | GO_IDLE_STATE        | Y | Y | Y | Y |   |   |   |   |                  |
| CMD1  | SEND_OP_COND         | Y | Y | Y | Y |   |   |   |   |                  |
| CMD2  | ALL_SEND_CID         | Y | Y | Y | Y |   |   |   |   |                  |
| CMD3  | SET_RELATIVE_ADDR    | Y | Y | Y | Y |   |   |   |   |                  |
| CMD4  | SET_DSR              | Y | Y | Y | Y |   |   |   |   |                  |
| CMD7  | SELECT/DSELECT_CARD  | Y | Y | Y | Y |   |   |   | Y |                  |
| CMD9  | SEND_CID             | Y | Y | Y | Y |   |   |   |   |                  |
| CMD10 | SEND_CSD             | Y | Y | Y | Y |   |   |   |   |                  |
| CMD11 | READ_DAT_UNTIL_STOP  | Y | Y | Y | Y |   | Y | Y |   |                  |
| CMD12 | STOP_TRANSMISSION    | Y | Y | Y | Y |   |   |   | Y |                  |
| CMD13 | SEND_STATUS          | Y | Y | Y | Y |   |   |   |   |                  |
| CMD15 | GO_INACTIVE_STATE    | Y | Y | Y | Y |   |   |   |   |                  |
| CMD16 | SET_BLOCKLEN         | Y | Y | Y | Y |   |   |   |   |                  |
| CMD17 | READ_SINGLE_BLOCK    | Y | Y | Y | Y | Y |   |   |   |                  |
| CMD18 | READ_MULTIPLE_BLOCK  | Y | Y | Y | Y | Y | Y |   |   | Open-ended       |
| CMD18 | READ_MULTIPLE_BLOCK  | Y | Y | Y | Y | Y |   |   |   | Predefine blocks |
| CMD20 | WRITE_DAT_UNTIL_STOP | Y | Y | Y | Y |   | Y | Y | Y |                  |
| CMD23 | SET_BLOCK_COUNT      | Y | Y | Y | Y |   |   |   |   |                  |
| CMD24 | WRITE_SINGLE_BLOCK   | Y | Y | Y | Y | Y |   |   | Y |                  |
| CMD25 | WRITE_MULTIPLE_BLOCK | Y | Y | Y | Y | Y | Y |   | Y | Open-ended       |
| CMD25 | WRITE_MULTIPLE_BLOCK | Y | Y | Y | Y | Y |   |   | Y | Predefine blocks |
| CMD26 | PROGRAM_CID          | Y | Y | Y | Y | Y |   |   | Y |                  |
| CMD27 | PROGRAM_CSD          | Y | Y | Y | Y | Y |   |   | Y |                  |
| CMD28 | SET_WRITE_PROT       | Y | Y | Y | Y |   |   |   | Y |                  |
| CMD29 | CLR_WRITE_PROT       | Y | Y | Y | Y |   |   |   | Y |                  |
| CMD30 | SEND_WRITE_PROT      | Y | Y | Y | Y | Y |   |   |   |                  |
| CMD32 | ERASE_WR_BLOCK_START | Y | Y | Y | Y |   |   |   |   |                  |
| CMD33 | ERASE_WR_BLOCK_END   | Y | Y | Y | Y |   |   |   |   |                  |
| CMD35 | ERASE_GROUP_START    | Y | Y | Y | Y |   |   |   |   |                  |
| CMD36 | ERASE_GROUP_END      | Y | Y | Y | Y |   |   |   |   |                  |

|        |                     |   |   |   |   |   |  |  |   |       |
|--------|---------------------|---|---|---|---|---|--|--|---|-------|
| CMD38  | ERASE               | Y | Y | Y | Y |   |  |  | Y |       |
| CMD39  | FAST_IO             | Y | Y | Y | Y |   |  |  |   |       |
| CMD40  | GO_IRQ_STATE        | Y | Y | Y | Y |   |  |  |   |       |
| CMD42  | LOCK/UNLOCK         | Y | Y | Y | Y | Y |  |  | Y |       |
| CMD55  | APP_CMD             | Y | Y | Y | Y |   |  |  |   |       |
| CMD56  | GEN_CMD             | Y | Y | Y | Y | Y |  |  |   | Read  |
| CMD56  | GEN_CMD             | Y | Y | Y | Y | Y |  |  | Y | Write |
| ACMD6  | SET_BUS_WIDTH       | Y | Y | Y | Y |   |  |  |   |       |
| ACMD13 | SD_STATUS           | Y | Y | Y | Y | Y |  |  |   |       |
| ACMD22 | SEND_NUM_WR_BLOCKS  | Y | Y | Y | Y |   |  |  |   |       |
| ACMD23 | SET_WR_BLOCK_COUNT  | Y | Y | Y | Y |   |  |  |   |       |
| ACMD41 | SD_SEND_OP_COND     | Y | Y | Y | Y |   |  |  |   |       |
| ACMD42 | SET_CLR_CARD_DETECT | Y | Y | Y | Y |   |  |  |   |       |
| ACMD51 | SEND_SCR            | Y | Y | Y | Y | Y |  |  |   |       |

**NOTE:** For stream read/write, STOP\_CMD is sent after finishing data transfer. For write, STOP\_CMD is with the last six bytes. For read, STOP\_CMD is sent after receiving data and card sends some data which MSC ignores.

## 33 OTG Controller

### 33.1 Overview

This chapter describes the USB On-The-Go (OTG) implemented in the processor.

The Universal Serial Bus (USB) supports serial data exchanges between a host computer and a variety of simultaneously accessible portable peripherals. Many of these portable devices would benefit a lot from being able to communicate to each other over the USB interface. And OTG make this possible. An OTG device can plays the role of both host and device.

Familiarity with the *Universal Serial Bus Specification*, Revision 1.1 and OTG supplement are necessary to fully understand the material contained in this section.

Features:

- Complies with the USB 2.0 standard for high-speed (480 Mbps) functions and with the *On-The-Go* supplement to the USB 2.0 specification
- Operates either as the function controller of a high- /full-speed USB peripheral or as the host/peripheral in point-to-point or multi-point communications with other USB functions
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- UTMI+ Level 3 Transceiver Interface
- Soft connect/disconnect
- 16 Endpoints:
- Supports control, interrupt, ISO and bulk transfer

### 33.2 Pin Description

**Table 33-1 OTG Pins Description**

| Name         | Type  | Description        |
|--------------|-------|--------------------|
| DP           | Inout | Data Positive      |
| DM           | Inout | Data Minus         |
| ID           | Inout | Identification     |
| DRVVBUS_PE10 | Out   | Charge pump enable |

### 33.3 Control and Status Overview

Your application controls the DWC\_otg core by reading from and writing to the Control and Status Registers (CSRs) through the AHB Slave interface. These registers are 32 bits wide, the addresses are 32- bit block aligned and the base address is 0x13500000.

Only the Core Global, Data FIFO Access, and Host Port registers can be accessed in both Host and Device modes. When the DWC\_otg core is operating in one mode, either Device or Host, the application must not access registers from the other mode. If an illegal access occurs, a Mode Mismatch interrupt is generated and reflected in the Core Interrupt register (GINTSTS.ModeMis). When the core switches from one mode to another, the registers in the new mode must be reprogrammed as they would be after a power-on reset.

### 33.4 CSR Memory Map

The CSR address map is fixed and does not depend on the core's configuration (for example, how many endpoints are implemented). Host and Device mode registers occupy different addresses. All registers are implemented in the AHB Clock domain.

Following figure shows the CSR address map.

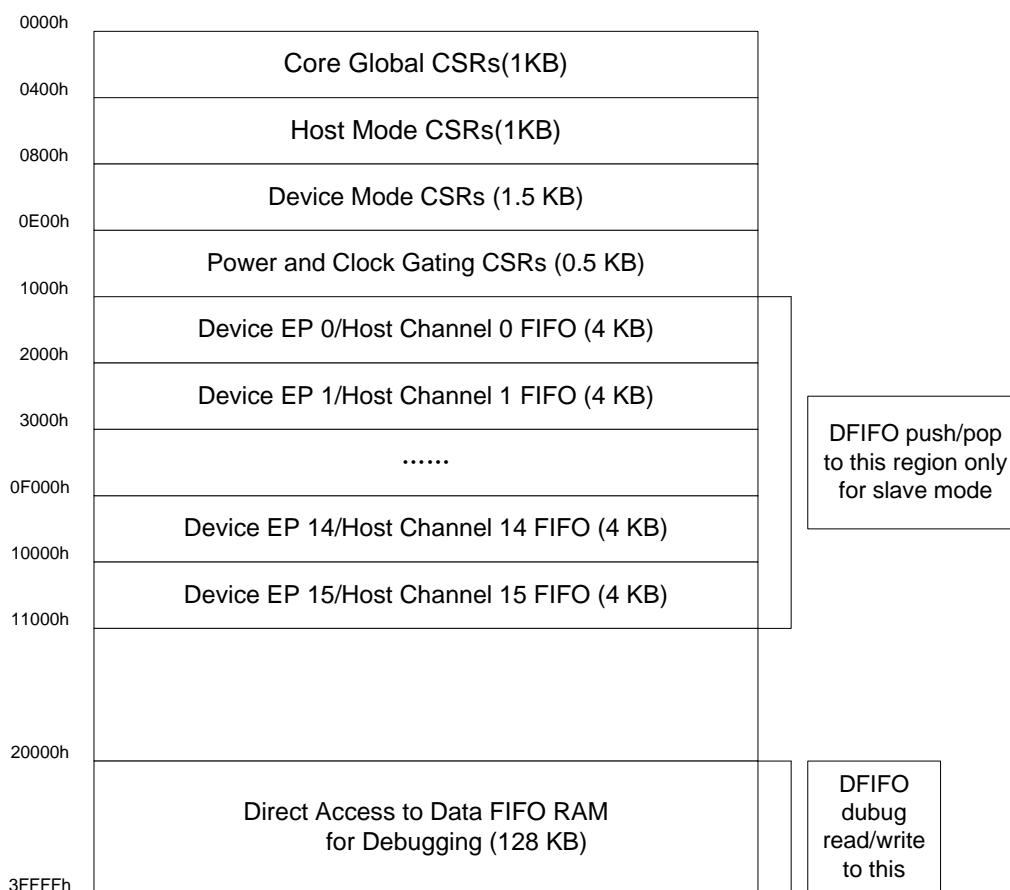


Figure 33-1 OTG CSR Memory Map

### 33.4.1 Register Maps

The tables in this section provide high-level summaries of each register and register group.

|                       |                                                                                                                                                                                                                                                                                         |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Register Name</b>  | Name of register types and register names ordered by offset address                                                                                                                                                                                                                     |
| <b>Acronym</b>        | <p>Shorthand names for registers that are mapped to the offset address. These are used extensively in the programming examples in the Programming Guide.</p> <p>The first letter is a prefix for the register type:</p> <p>G: Core Global</p> <p>H: Host mode</p> <p>D: Device mode</p> |
| <b>Offset Address</b> | Address, in hexadecimal (h), of the first byte of each register.                                                                                                                                                                                                                        |

**Note:** FIFO size and FIFO depth are used interchangeably.

### 33.4.2 Global CSR Map

These registers are available in both Host and Device modes.

**Table 33-2 Core Global CSR Map (000h-3FFh)**

| Acronym   | Offset Address | Register Name                                                               |
|-----------|----------------|-----------------------------------------------------------------------------|
| GOTGCTL   | 000h           | "Control and Status Register (GOTGCTL)"                                     |
| GOTGINT   | 004h           | "Interrupt Register (GOTGINT)"                                              |
| GAHBCFG   | 008h           | "AHB Configuration Register (GAHBCFG)"                                      |
| GUSBCFG   | 00Ch           | "USB Configuration Register (GUSBCFG)"                                      |
| GRSTCTL   | 010h           | "Reset Register (GRSTCTL)"                                                  |
| GINTSTS   | 014h           | "Interrupt Register (GINTSTS)"                                              |
| GINTMSK   | 018h           | "Interrupt Mask Register (GINTMSK)"                                         |
| GRXSTSR   | 01Ch           | "Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)" |
| GRXSTSP   | 020h           |                                                                             |
| GRXFSIZ   | 024h           | "Receive FIFO Size Register (GRXFSIZ)"                                      |
| GNPTXFSIZ | 028h           | "Non-Periodic Transmit FIFO Size Register (GNPTXFSIZ)"                      |
| GNPTXSTS  | 02Ch           | "Non-Periodic Transmit FIFO/Queue Status Register (GNPTXSTS)"               |
|           | 030h - 038h    | Reserved                                                                    |
| GUID      | 03Ch           | "User ID Register (GUID)"                                                   |
| GSNPSID   | 040h           | "Synopsys ID Register (GSNPSID)"                                            |
| GHWCFG1   | 044h           | "User HW Config1 Register (GHWCFG1)"                                        |
| GHWCFG2   | 048h           | "User HW Config2 Register (GHWCFG2)"                                        |
| GHWCFG3   | 04Ch           | "User HW Config3 Register (GHWCFG3)"                                        |
| GHWCFG4   | 050h           | "User HW Config4 Register (GHWCFG4)"                                        |
| GLPMCFCG  | 054h           | "Core LPM Configuration Register (GLPMCFCG)"                                |

|           |                             |                                                              |
|-----------|-----------------------------|--------------------------------------------------------------|
|           | 0x58h                       | Reserved                                                     |
| GDFIFOCFG | 05Ch                        | "DFIFO Software Config Register (GDFIFOCFG)"                 |
| GADPCTL   | 0x60h                       | "ADP Timer, Control and Status Register (GADPCTL)"           |
| HPTXFSIZ  | 100h                        | "Host Periodic Transmit FIFO Size Register (HPTXFSIZ)"       |
| DPTXFSIZn | 104h-13Ch<br>Dedicated FIFO | "Device Periodic Transmit FIFO-n Size Register (DPTXFSIZn)"  |
| DIEPTXFn  | 104h-13Ch<br>Dedicated FIFO | "Device IN Endpoint Transmit FIFO Size Register: (DIEPTXFn)" |
|           | 140h-3FFh                   | Reserved                                                     |

### 33.4.3 Host Mode CSR Map

These registers must be programmed every time the core changes to Host mode.

**Table 33-3 Host Mode CSR Map (400h-7FFh)**

| Acronym   | Offset Address | Register Name                                               |
|-----------|----------------|-------------------------------------------------------------|
| HCFG      | 400h           | Host Configuration Register (HCFG)                          |
| HFIR      | 404h           | Host Frame Interval Register (HFIR)                         |
| HFNUM     | 408h           | Host Frame Number/Frame Time Remaining Register (HFNUM)     |
|           | 40Ch           | Reserved                                                    |
| HPTXSTS   | 410h           | Host Periodic Transmit FIFO/Queue Status Register (HPTXSTS) |
| HAIN      | 414h           | Host All Channels Interrupt Register (HAIN)                 |
| HAINMSK   | 418h           | Host All Channels Interrupt Mask Register (HAINMSK)         |
| HFLBAddr  | 41Ch           | Host Frame List Base Address Register (HFLBAddr)            |
| HPRT      | 440h           | Host Port Control and Status Register (HPRT)                |
|           | 444h-4FCh      | Reserved                                                    |
| HCCHARn   | 500h - 6E0     | Host Channel-n Characteristics Register (HCCHARn)           |
| HCSPLTn   | 504h - 6E4     | Host Channel-n Split Control Register (HCSPLTn)             |
| HCINTn    | 508h - 6E8     | Host Channel-n Interrupt Register (HCINTn)                  |
| HCINTMSKn | 50Ch - 6EC     | Host Channel-n Interrupt Mask Register (HCINTMSKn)          |
| HCTSIZn   | 510h - 6F0     | Host Channel-n Transfer Size Register (HCTSIZn)             |
|           | 518h - 6F8     | Reserved                                                    |
| HCDMABn   | 51Ch - 6FCh    | Host Channel-n DMA Buffer Address Register (HCDMABn)        |
|           | 6F0-7FFh       | Reserved                                                    |

### 33.4.4 Device Mode CSR Map

These registers must be programmed every time the core changes to Device mode.

**Table 33-4 Device Mode CSR Map (800h-BFFh)**

| Acronym                  | Offset Address | Register Name                                                                      |
|--------------------------|----------------|------------------------------------------------------------------------------------|
|                          | 800h-ACh       | Device Logical IN Endpoint-Specific Registers                                      |
| DCFG                     | 800h           | Device Configuration Register (DCFG)                                               |
| DCTL                     | 804h           | Device Control Register (DCTL)                                                     |
| DSTS                     | 808h           | Device Status Register (DSTS)                                                      |
|                          | 80Ch           | Reserved                                                                           |
| DIEPMSK                  | 810h           | Device IN Endpoint Common Interrupt Mask Register (DIEPMSK)                        |
| DOEPMSK                  | 814h           | Device OUT Endpoint Common Interrupt Mask Register (DOEPMSK)                       |
| DAINT                    | 818h           | Device All Endpoints Interrupt Register (DAINT)                                    |
| DAINTMSK                 | 81Ch           | Device All Endpoints Interrupt Mask Register (DAINTMSK)                            |
| DTKNQR1                  | 820h           | Device IN Token Sequence Learning Queue Read Register 1 (DTKNQR1)                  |
| DTKNQR2                  | 824h           | Device IN Token Sequence Learning Queue Read Register 2 (DTKNQR2)                  |
| DTKNQR3                  | 830h           | Device IN Token Sequence Learning Queue Read Register 3 (DTKNQR3)                  |
| DTKNQR4                  | 834h           | Device IN Token Sequence Learning Queue Read Register 4 (DTKNQR4)                  |
| DVBUSDIS                 | 828h           | Device VBUS Discharge Time Register (DVBUSDIS)                                     |
| DVBUSPULSE               | 82Ch           | Device VBUS Pulsing Time Register (DVBUSPULSE)                                     |
| DTHRCTL                  | 830h           | Device Threshold Control Register (DTHRCTL)                                        |
| DIEPEMPMSK               | 834h           | Device IN Endpoint FIFO Empty Interrupt Mask Register: (DIEPEMPMSK)                |
| DEACHINT                 | 838h           | Device Each Endpoint Interrupt Register (DEACHINT)                                 |
| DEACHINTMSK              | 83Ch           | Device Each Endpoint Interrupt Register Mask (DEACHINTMSK)                         |
| DIEPEACHMSK <sub>n</sub> | 840h           | Device Each In Endpoint-n Interrupt Register (DIEPEACHMSK <sub>n</sub> )           |
| DOEPEACHMSK <sub>n</sub> | 880h           | Device Each Out Endpoint-n Interrupt Register (DOEPEACHMSK <sub>n</sub> )          |
| DIEPCTL0                 | 900h           | Device Control IN Endpoint 0 Control Register (DIEPCTL0)                           |
|                          | 904h           | Reserved                                                                           |
| DIEPCTL <sub>n</sub>     | 920h-AE0h      | Device Endpoint-n Control Register (DIEPCTL <sub>n</sub> /DOEPCTL <sub>n</sub> )   |
| DIEPINT <sub>n</sub>     |                | Device Endpoint-n Interrupt Register (DIEPINT <sub>n</sub> /DOEPINT <sub>n</sub> ) |
|                          | 90Ch           | Reserved                                                                           |

|                                    |           |                                                                                                |
|------------------------------------|-----------|------------------------------------------------------------------------------------------------|
| DIEPTSIZE0                         | 910h      | Device Endpoint 0 Transfer Size Register<br>(DIEPTSIZE0/DOEPTSIZE0)                            |
| DIEPTSIZE <sub>n</sub>             | 910h      | Device Endpoint-n Transfer Size Register<br>(DIEPTSIZE <sub>n</sub> /DOEPTSIZE <sub>n</sub> )  |
| DIEPDMA <sub>n</sub>               | 914h      | Device Endpoint-n DMA Address Register<br>(DIEPDMA <sub>n</sub> /DOEPDMA <sub>n</sub> )        |
| DTXFSTSn                           | 918h      | Device IN Endpoint Transmit FIFO Status Register<br>(DTXFSTSn)                                 |
| DIEPDMA0/DIEP<br>DMAB <sub>n</sub> | 91Ch      | Device Endpoint-n DMA Buffer Address Register<br>(DIEPDMA <sub>n</sub> /DOEPDMA <sub>n</sub> ) |
| DOEPCTL0                           | B00h      | Device Control OUT Endpoint 0 Control Register<br>(DOEPCTL0)                                   |
|                                    | B04h      | Reserved                                                                                       |
| DOEPCTL <sub>n</sub>               | B20h-CE0h | Device Endpoint-n Control Register<br>(DIEPCTL <sub>n</sub> /DOEPCTL <sub>n</sub> )            |
| DOEPINT <sub>n</sub>               | B08h      | Device Endpoint-n Interrupt Register<br>(DIEPINT <sub>n</sub> /DOEPINT <sub>n</sub> )          |
|                                    | B0Ch      | Reserved                                                                                       |
| DOEPTSIZE0                         | B10h      | Device Endpoint 0 Transfer Size Register<br>(DIEPTSIZE0/DOEPTSIZE0)                            |
| DOEPTSIZE <sub>n</sub>             | B30h-DB0h | Device Endpoint-n Transfer Size Register<br>(DIEPTSIZE <sub>n</sub> /DOEPTSIZE <sub>n</sub> )  |
| DOEPDMA <sub>n</sub>               | B14h-CF4h | Device Endpoint-n DMA Address Register<br>(DIEPDMA <sub>n</sub> /DOEPDMA <sub>n</sub> )        |
| DOEPDMA0/<br>DOEPDMAB <sub>n</sub> | B1Ch-CFCh | Device Endpoint-n DMA Buffer Address Register<br>(DIEPDMA <sub>n</sub> /DOEPDMA <sub>n</sub> ) |

### 33.4.5 Data FIFO (DFIFO) Access Register Map

These registers, available in both Host and Device modes, are used to read or write the FIFO space for a specific endpoint or a channel, in a given direction. If a host channel is of type IN, the FIFO can only be read on the channel. Similarly, if a host channel is of type OUT, the FIFO can only be written on the channel.

**Table 33-5 Data FIFO (DFIFO) Access Register Map**

| FIFO Access Register Section                                                                                              | Address Range | Access |
|---------------------------------------------------------------------------------------------------------------------------|---------------|--------|
| Device IN Endpoint 0/Host OUT Channel 0: DFIFO Write Access<br>Device OUT Endpoint 0/Host IN Channel 0: DFIFO Read Access | 1000h-1FFCh   | WO/RO  |
| Device IN Endpoint 1/Host OUT Channel 1: DFIFO Write Access<br>Device OUT Endpoint 1/Host IN Channel 1: DFIFO Read Access | 2000h-2FFCh   | WO/RO  |
| ...                                                                                                                       | ...           | ...    |



|                                                                                                                               |               |       |
|-------------------------------------------------------------------------------------------------------------------------------|---------------|-------|
| Device IN Endpoint 14/Host OUT Channel 14: DFIFO Write Access<br>Device OUT Endpoint 14/Host IN Channel 14: DFIFO Read Access | F000h-FFFCh   | WO/RO |
| Device IN Endpoint 15/Host OUT Channel 15: DFIFO Write Access<br>Device OUT Endpoint 15/Host IN Channel 15: DFIFO Read Access | 10000h-10FFCh | WO/RO |

### 33.4.6 Interrupt Hierarchy

Following figure displays the DWC\_otg interrupt hierarchy.

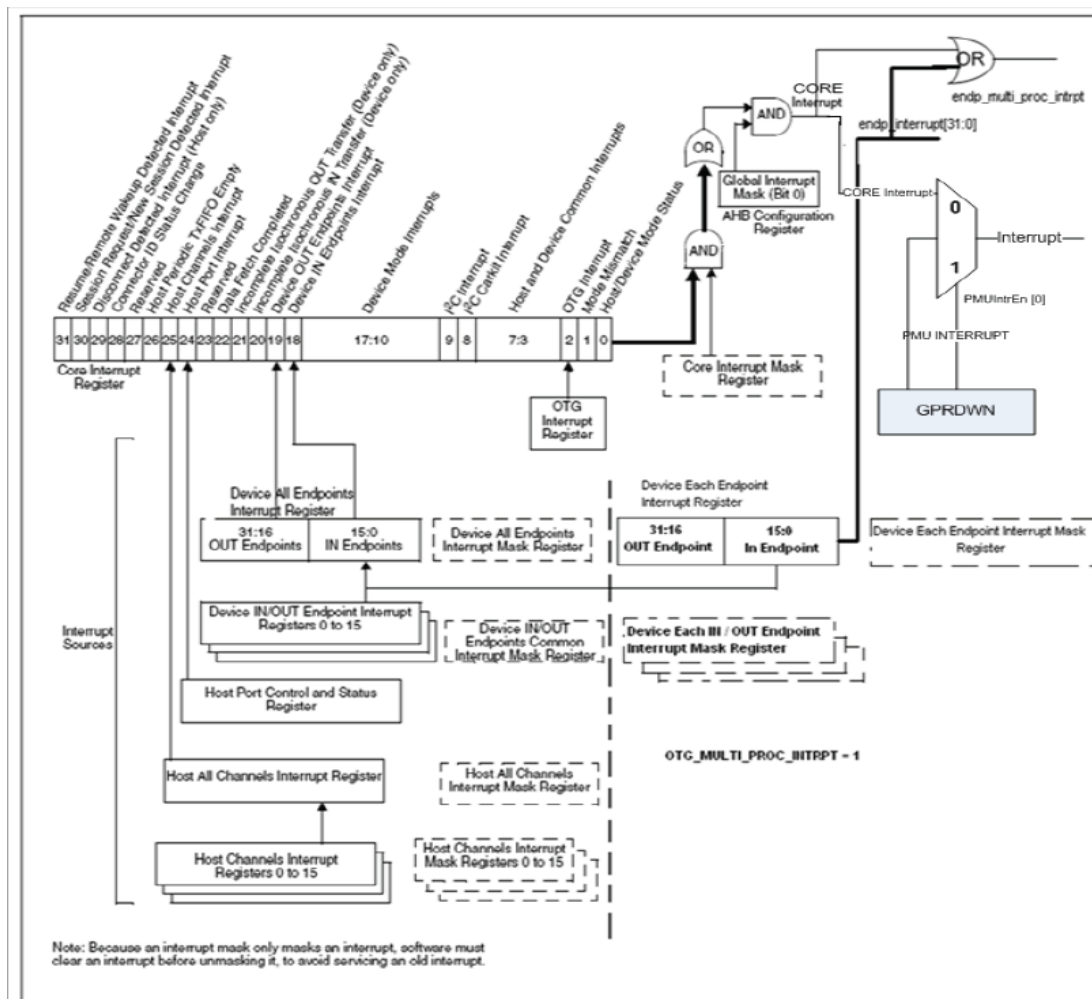


Figure 33-2 Interrupt Hierarchy

### 33.4.6.1 The Core Interrupt Handler

Following figure illustrates how the core interrupt handler works.

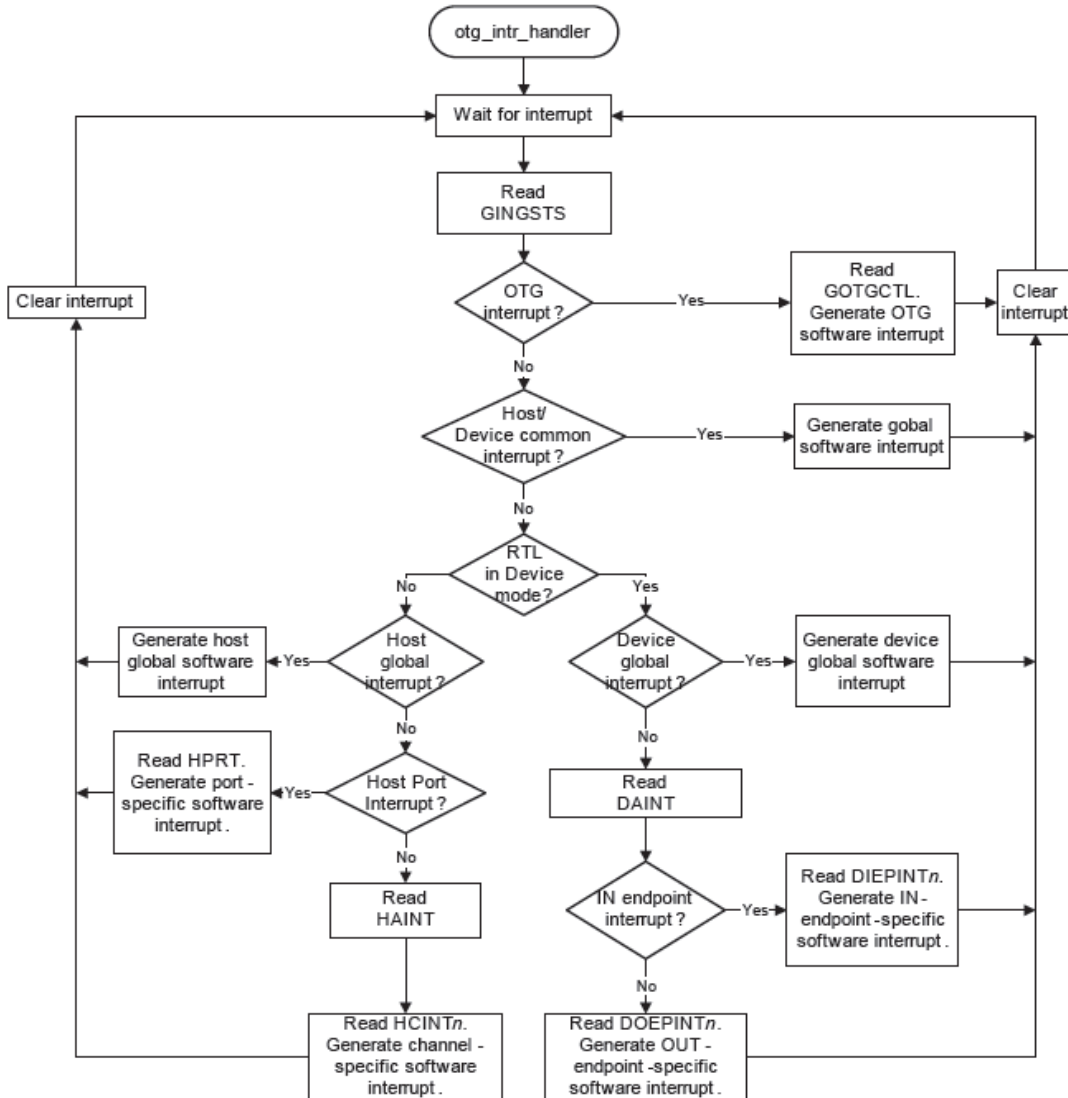


Figure 33-3 Core Interrupt Handler

## 33.5 Register Descriptions

This section describes Core Global, Device Mode, Host Mode CSRs. Detailed register programming examples are provided in the Programming Guide.

### 33.5.1 Application Access to the CSRs

The Access column of each register description that follows specifies how the application and the core can access the register fields of the CSRs. The following conventions are used:

|                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Read Only (RO)</b>                                             | Register field can only be read by the application. Writes to read-only fields have no effect.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Write Only (WO)</b>                                            | Register field can only be written by the application.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Read and Write (R_W)</b>                                       | Register field can be read and written by the application. The application can set this field by writing 1'b1 and can clear it by writing 1'b0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Read, Write, and Self Clear (R_W_SC)</b>                       | Register field can be read and written by the application (Read and Write), and is cleared to 1'b0 by the core (Self Clear). The conditions under which the core clears this field are explained in detail in the field's description.                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Read, Write, Self Set, and Self Clear (R_W_SS_SC)</b>          | Register field can be read and written by the application (Read and Write), set to 1'b1 by the core on certain USB events (Self Set), and cleared to 1'b0 by the core (Self Clear). The conditions under which the core sets and clears this field are explained in the field's description. (Only the Port Resume bit of the Host Port Control and Status register, HPRT.PrtRes, uses this access type).                                                                                                                                                                                                                                                      |
| <b>Read, Self Set, and Write Clear (R_SS_WC)</b>                  | Register field can be read by the application (Read), can be set to 1'b1 by the core on a certain internal or USB or AHB event (Self Set), and can be cleared to 1'b0 by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 has no effect on this field. The conditions under which the core sets this field are explained in detail in the field's description. (For example, interrupt bits.)                                                                                                                                                                                                                             |
| <b>Read, Write Set, and Self Clear (R_WS_SC)</b>                  | Register field can be read by the application (Read), can be set to 1'b1 by the application with a register write of 1'b1 (Write Set), and is cleared to 1'b0 by the core (Self Clear). The application cannot clear this type of field, and a register write of 1'b0 to this bit has no effect on this field. The conditions under which the core clears this field are explained in detail in the field's description. (For example, reset signals)                                                                                                                                                                                                          |
| <b>Read, Self Set, and Self Clear or Write Clear (R_SS_SC_WC)</b> | Register field can be read by the application (Read), can be set to 1'b1 by the core on certain internal or USB or AHB events (Self Set), and can be cleared to 1'b0 either by the core itself (Self Clear) or by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 to this bit has to no effect on this field.<br><br>The conditions under which the core sets or clears this field are explained in the field's description. (Only the Port Enable bit of the Host Port Control and Status register, HPRT.PrtEna, and the VStatus Done bit of the PHY Vendor Control register, GPVNDCTL.VStsDone, use this access type.) |

**Note:** Always program Reserved fields with 0s. Treat read values from Reserved fields as unknowns(Xs).

### 33.5.2 Overview of Commonly Used Register Bits

This section provides an overview of the commonly used registers and bits. For a complete description of all the registers, see the corresponding sections.

Table 33-6 List of Commonly Used Register Bits

| Bit Number | Register/Bit Name                       | Description                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | "Control and Status Register (GOTGCTL)" | The OTG Control and Status register controls the behavior and reflects the status of the OTG function of the core.                                                                                                                                                                                                                                                                                                  |
| 11         | Device HNP Enabled (DevHNPEn)           | The application sets this bit when it successfully receives a SetFeature.SetHNPEnable command from the connected USB host. <ul style="list-style-type: none"> <li>▪ 1'b0: HNP is not enabled in the application</li> <li>▪ 1'b1: HNP is enabled in the application</li> </ul>                                                                                                                                       |
| 10         | Host Set HNP Enable (HstSetHNPEn)       | The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. <ul style="list-style-type: none"> <li>▪ 1'b0: Host Set HNP is not enabled</li> <li>▪ 1'b1: Host Set HNP is enabled</li> </ul>                                                                                                                                              |
|            | "AHB Configuration Register (GAHBCFG)"  | This register can be used to configure the core after power-on or a change in mode. This register mainly contains AHB system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the AHB or the USB.                                                                                 |
| 5          | DMA Enable (DMAEn)                      | <ul style="list-style-type: none"> <li>▪ 1'b0: Core operates in Slave mode</li> <li>▪ 1'b1: Core operates in a DMA mode</li> </ul>                                                                                                                                                                                                                                                                                  |
|            | "USB Configuration Register (GUSBCFG)"  | This register can be used to configure the core after power-on or a changing to Host mode or Device mode. It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the AHB or the USB. Do not make changes to this register after the initial programming.                                                               |
| 30         | Force Device Mode (ForceDevMode)        | Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. <ul style="list-style-type: none"> <li>▪ 1'b0: Normal Mode</li> <li>▪ 1'b1: Force Device Mode</li> </ul> <p>After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 <math>\mu</math>s is sufficient.</p> |
| 29         | Force Host Mode (ForceHstMode)          | Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. <ul style="list-style-type: none"> <li>▪ 1'b0: Normal Mode</li> <li>▪ 1'b1: Force Host Mode</li> </ul> <p>After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale</p>                                                                 |

|    |                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                                                        | down mode, waiting for 500 $\mu$ s is sufficient.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 9  | HNP-Capable (HNPCap)                                                   | <p>The application uses this bit to control the DWC_otg core's HNP capabilities.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: HNP capability is not enabled.</li> <li>▪ 1'b1: HNP capability is enabled.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 8  | SRP-Capable (SRPCap)                                                   | <p>The application uses this bit to control the DWC_otg core SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: SRP capability is not enabled.</li> <li>▪ 1'b1: SRP capability is enabled.</li> </ul>                                                                                                                                                                                                                                                                                                                       |
| 6  | USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select | <p>The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY</li> <li>▪ 1'b1: USB 1.1 full-speed serial transceiver</li> </ul> <p>This bit is always 0, with Read Only access.</p>                                                                                                                                                                                                                                                                                                                                                            |
| 4  | ULPI or UTMI+ Select (ULPI_UTMI_Sel)                                   | <p>The application uses this bit to select either a UTMI+ interface or ULPI Interface.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: UTMI+ Interface</li> <li>▪ 1'b1: ULPI Interface</li> </ul> <p>This bit is always 0, with Read Only access.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 3  | PHY Interface (PHYIf)                                                  | <p>The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface.</p> <p>1'b0: 8 bits<br/>1'b1: 16 bits</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|    | "Host Configuration Register (HCFG)"                                   | This register configures the core after power-on. Do not make changes to this register after initializing the host.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23 | Enable Scatter/gather DMA in Host mode (DescDMA)                       | <p>When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation.</p> <p>NOTE: This bit must be modified only once after a reset. The following combinations are available for programming:</p> <ul style="list-style-type: none"> <li>▪ GAHBCFG.DMAEn=0, HCFG.DescDMA=0 =&gt; Slave mode</li> <li>▪ GAHBCFG.DMAEn=0, HCFG.DescDMA=1 =&gt; Invalid</li> <li>▪ GAHBCFG.DMAEn=1, HCFG.DescDMA=0 =&gt; Buffered DMA mode</li> <li>▪ GAHBCFG.DMAEn=1, HCFG.DescDMA=1 =&gt; Scatter/Gather DMA mode</li> </ul> <p>In non Scatter/Gather DMA mode, this bit is reserved.</p> |
| 2  | FS- and LS-Only                                                        | The application uses this bit to control the core's enumeration                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|       |                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | Support(FSLSSupp)                                   | <p>speed. Using this bit, the application can make the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device</li> <li>▪ 1'b1: FS/LS-only, even if the connected device can support HS</li> </ul>                                                                           |
|       | "Host Channel-n Characteristics Register (HCCHARn)" |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 19:18 | Endpoint Type (EPTYPE)                              | <p>Indicates the transfer type selected.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: Control</li> <li>▪ 2'b01: Isochronous</li> <li>▪ 2'b10: Bulk</li> <li>▪ 2'b11: Interrupt</li> </ul>                                                                                                                                                                                                                                                                                           |
| 15    | Endpoint Direction (EPDir)                          | <p>Indicates whether the transaction is IN or OUT.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: OUT</li> <li>▪ 1'b1: IN</li> </ul>                                                                                                                                                                                                                                                                                                                                                   |
| 14:11 | Endpoint Number (EPNum)                             | Indicates the endpoint number on the device serving as the data source or sink.                                                                                                                                                                                                                                                                                                                                                                                                            |
| 10:0  | Maximum Packet Size (MPS)                           | Indicates the maximum packet size of the associated endpoint.                                                                                                                                                                                                                                                                                                                                                                                                                              |
|       | "Host Channel-n Transfer Size Register (HCTSIZn)"   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 30:29 | PID (Pid)                                           | <p>The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: DATA0</li> <li>▪ 2'b01: DATA2</li> <li>▪ 2'b10: DATA1</li> <li>▪ 2'b11: MDATA (non-control)</li> </ul>                                                                                                                                                                     |
|       | "Device Configuration Register (DCFG)"              | <p>This register configures the core in Device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.</p>                                                                                                                                                                                                                                                                                                   |
| 25:24 | Periodic Scheduling Interval (PerSchIntvl)          | <p>PerSchIntvl must be programmed only for Scatter/Gather DMA mode. Description: This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)frame.</p> <ul style="list-style-type: none"> <li>▪ When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching</li> </ul> |

|      |                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      |                                                          | <p>periodic IN endpoint data.</p> <ul style="list-style-type: none"> <li>▪ When no periodic endpoints are active, then the internal DMA engine services non-periodic endpoints, ignoring this field.</li> <li>▪ After the specified time within a (micro)frame, the DMA switches to fetching for non-periodic endpoints.</li> <li>▪ 2'b00: 25% of (micro)frame.</li> <li>▪ 2'b01: 50% of (micro)frame.</li> <li>▪ 2'b10: 75% of (micro)frame.</li> <li>▪ 2'b11: Reserved.</li> </ul>                                                                                                                                                                       |
| 23   | Enable Scatter/Gather DMA in Device mode (DescDMA)       | <p>When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation.</p> <p>NOTE: This bit must be modified only once after a reset.</p> <p>The following combinations are available for programming:</p> <ul style="list-style-type: none"> <li>▪ GAHBCFG.DMAEn=0,DCFG.DescDMA=0 =&gt; Slave mode</li> <li>▪ GAHBCFG.DMAEn=0,DCFG.DescDMA=1 =&gt; Invalid</li> <li>▪ GAHBCFG.DMAEn=1,DCFG.DescDMA=0 =&gt; Buffered DMA mode</li> <li>▪ GAHBCFG.DMAEn=1,DCFG.DescDMA=1 =&gt; Scatter/Gather DMA mode</li> </ul>                              |
| 10:4 | Device Address (DevAddr)                                 | The application must program this field after every SetAddress control command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 1:0  | Device Speed (DevSpd)                                    | <p>Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. See "Device Initialization" in the Programming Guide for details.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz)</li> <li>▪ 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz)</li> <li>▪ 2'b10: Reserved</li> <li>▪ 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz)</li> </ul> |
|      | "Device Endpoint-n Control Register (DIEPCTLn/DOEPCTLn)" | The application uses this register to control the behavior of each logical endpoint other than endpoint 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 29   | Set DATA1 PID (SetD1PID)                                 | <p>Applies to interrupt/bulk IN and OUT endpoints only.</p> <p>Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.</p> <p>This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.</p>                                                                                                                                                                                                                                                                                                                                                                                              |
| 28   | Set DATA0 PID (SetD0PID)                                 | <p>Applies to interrupt/bulk IN and OUT endpoints only.</p> <p>Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

|       |                           |                                                                                                                                                                                                                                                       |
|-------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |                           | This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.                                                                                                                                                            |
| 19:18 | Endpoint Type (EPTYPE)    | Applies to IN and OUT endpoints.<br>This is the transfer type supported by this logical endpoint. <ul style="list-style-type: none"> <li>▪ 2'b00: Control</li> <li>▪ 2'b01: Isochronous</li> <li>▪ 2'b10: Bulk</li> <li>▪ 2'b11: Interrupt</li> </ul> |
| 10:00 | Maximum Packet Size (MPS) | Applies to IN and OUT endpoints.<br>The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.                                                                                    |

### 33.5.3 Global Registers

These registers are available in both Host and Device modes, and do not need to be reprogrammed when switching between these modes.

#### 33.5.3.1 Control and Status Register (GOTGCTL)

◇ Offset: 000h

The OTG Control and Status register controls the behavior and reflects the status of the OTG function of the core.

**Table 33-7 Control and Status Register: GOTGCTL**

| Field | Description                                                                                                                                                                                                                                                                                                                                           | Mode            | Reset                                                                             | Access |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-----------------------------------------------------------------------------------|--------|
| 31:26 | Reserved                                                                                                                                                                                                                                                                                                                                              | Host and Device |                                                                                   | RO     |
| 27    | Chirp On Enable (ChirpEn)<br>This bit when programmed to 1'b1 results in the core asserting chirp_on before sending an actual Chirp "K" signal on USB. This bit is present only if OTG_BC_SUPPORT = 1. If OTG_BC_SUPPORT != 1, this bit is a reserved bit.                                                                                            | Device only     | 1'b0                                                                              | RW     |
| 26:22 | Multi Valued ID pin (MultValIdBc)<br>Battery Charger ACA inputs in the following order: <ul style="list-style-type: none"> <li>▪ Bit 26 - rid_float.</li> <li>▪ Bit 25 - rid_gnd</li> <li>▪ Bit 24 - rid_a</li> <li>▪ Bit 23 - rid_b</li> <li>▪ Bit 22 - rid_c</li> </ul> These bits are valid only if OTG_BC_SUPPORT=1, otherwise they are reserved. | Host and Device | Configurable.<br>The reset value is unknown if OTG_BC_SUPPORT =1 and 0 otherwise. | RO     |
| 21    | Reserved                                                                                                                                                                                                                                                                                                                                              | Host and Device |                                                                                   | RO     |



|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Device          |              |     |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------|-----|
| 20    | OTG Version (OTGVer)<br>Indicates the OTG revision. <ul style="list-style-type: none"> <li>▪ 1'b0: OTG Version 1.3. In this version the core supports Data line pulsing and VBus pulsing for SRP.</li> <li>▪ 1'b1: OTG Version 2.0. In this version the core supports only Data line pulsing for SRP.</li> </ul>                                                                                                                                                                                                                                    |                 | 1'b0         | RW  |
| 19    | B-Session Valid (BSesVld)<br>Indicates the Device mode transceiver status. <ul style="list-style-type: none"> <li>▪ 1'b0: B-session is not valid.</li> <li>▪ 1'b1: B-session is valid.</li> </ul> In OTG mode, you can use this bit to determine if the device is connected or disconnected.<br><b>Note:</b> If you do not enable OTG features (such as SRP and HNP), the read reset value will be 1. The vbus assigns the values internally for non- SRP or non-HNP configurations.<br>In case of OTG_MODE=0, the reset value of this bit is 1'b0. | Device only     | Configurable | RO  |
| 18    | A-Session Valid (ASesVld)<br>Indicates the Host mode transceiver status. <ul style="list-style-type: none"> <li>▪ 1'b0: A-session is not valid</li> <li>▪ 1'b1: A-session is valid</li> </ul> <b>Note:</b> If you do not enable OTG features (such as SRP and HNP), the read reset value will be 1. The vbus assigns the values internally for non- SRP or non-HNP configurations.<br>In device mode, this bit is reserved.                                                                                                                         | Host only       | Configurable | RO  |
| 17    | Long/Short Debounce Time (DbncTime)<br>Indicates the debounce time of a detected connection. <ul style="list-style-type: none"> <li>▪ 1'b0: Long debounce time, used for physical connections (100 ms + 2.5 μs)</li> <li>▪ 1'b1: Short debounce time, used for soft connections (2.5 μs)</li> </ul>                                                                                                                                                                                                                                                 | Host only       | 1'b0         | RO  |
| 16    | Connector ID Status (ConIDSts)<br>Indicates the connector ID status on a connect event. <ul style="list-style-type: none"> <li>▪ 1'b0: The DWC_otg core is in A-Device mode</li> <li>▪ 1'b1: The DWC_otg core is in B-Device mode</li> </ul>                                                                                                                                                                                                                                                                                                        | Host and Device | 1'b1         | RO  |
| 15:12 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Host and Device |              | RO  |
| 11    | Device HNP Enabled (DevHNPEn)<br>The application sets this bit when it successfully receives                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Device Only     | 1'b0         | R_W |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                |      |     |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|------|-----|
|    | <p>a SetFeature.SetHNPEnable command from the connected USB host.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: HNP is not enabled in the application</li> <li>▪ 1'b1: HNP is enabled in the application</li> </ul>                                                                                                                                                                                                                                               | OTG configurations             |      |     |
| 10 | <p>Host Set HNP Enable (HstSetHNPEn)</p> <p>The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Host Set HNP is not enabled</li> <li>▪ 1'b1: Host Set HNP is enabled</li> </ul>                                                                                                                                                 | Device Only OTG configurations | 1'b0 | R_W |
| 9  | <p>HNP Request (HNPREq)</p> <p>The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: No HNP request</li> <li>▪ 1'b1: HNP request</li> </ul> | Device Only OTG configurations | 1'b0 | R_W |
| 8  | <p>Host Negotiation Success (HstNegScs)</p> <p>The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPREq) bit in this register is set.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Host negotiation failure</li> <li>▪ 1'b1: Host negotiation success</li> </ul>                                                                                                                                        | Device only                    | 1'b0 | RO  |
| 7  | <p>B-Peripheral Session Valid OverrideValue (BvalidOvVal)</p> <p>This bit is used to set the Override value for Bvalid signal when GOTGCTL.BvalidOvEn is set.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Bvalid value is 1'b0 when GOTGCTL.BvalidOvEn = 1.</li> <li>▪ 1'b1: Bvalid value is 1'b1 when GOTGCTL.BvalidOvEn = 1.</li> </ul>                                                                                                                       | Device only                    | 1'b0 | R_W |
| 6  | <p>B-Peripheral Session Valid Override Enable (BvalidOvEn)</p> <p>This bit is used to enable/disable the software to override the Bvalid signal using the GOTGCTL.BvalidOvVal.</p> <ul style="list-style-type: none"> <li>▪ 1'b1: Internally Bvalid received from the PHY is overridden with GOTGCTL.BvalidOvVal.</li> <li>▪ 1'b0: Override is disabled and Bvalid signal from the respective PHY selected is used internally by the core.</li> </ul>                  | Device only                    | 1'b0 | R_W |
| 5  | <p>A-Peripheral Session Valid OverrideValue (AvalidOvVal)</p> <p>This bit is used to set the Override value for Avalid signal when GOTGCTL.AvalidOvEn is set.</p>                                                                                                                                                                                                                                                                                                      | Host only                      | 1'b0 | R_W |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |      |     |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|------|-----|
|   | <ul style="list-style-type: none"> <li>▪ 1'b0: Avalid value is 1'b0 when GOTGCTL.AvalidOvEn = 1.</li> <li>▪ 1'b1: Avalid value is 1'b1 when GOTGCTL.AvalidOvEn = 1.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |      |     |
| 4 | <p>A-Peripheral Session Valid Override Enable (AvalidOvEn)</p> <p>This bit is used to enable/disable the software to override the Avalid signal using the GOTGCTL.AvalidOvVal.</p> <ul style="list-style-type: none"> <li>▪ 1'b1: Internally Avalid received from the PHY is overridden with GOTGCTL.AvalidOvVal.</li> <li>▪ 1'b0: Override is disabled and Avalid signal from the respective PHY is used internally by the core.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                        | Host only   | 1'b0 | R_W |
| 3 | <p>VBUS Valid Override Value (VbvalidOvVal)</p> <p>This bit is used to set the Override value for vbus valid signal when GOTGCTL.VbusvalidOvEn is set.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: vbusvalid value is 1'b0 when GOTGCTL.VbvalidOvEn = 1.</li> <li>▪ 1'b1: vbusvalid value is 1'b1 when GOTGCTL.VbvalidOvEn = 1.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Host only   | 1'b0 | R_W |
| 2 | <p>VBUS Valid Override Enable (VbvalidOvEn)</p> <p>This bit is used to enable/disable the software to override the vbus-valid signal using the GOTGCTL.vbvalidOvVal.</p> <ul style="list-style-type: none"> <li>▪ 1'b1: The vbus-valid signal received from the PHY is overridden with GOTGCTL.vbvalidOvVal.</li> <li>▪ 1'b0: Override is disabled and avalid signal from the respective PHY is used internally by the core.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                             | Host only   | 1'b0 | R_W |
| 1 | <p>Session Request (SesReq)</p> <p>The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared.</p> <p>If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: No session request</li> <li>▪ 1'b1: Session request</li> </ul> | Device only | 1'b0 | R_W |
| 0 | <p>Session Request Success (SesReqScs)</p> <p>The core sets this bit when a session request initiation is successful.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Device only | 1'b0 | RO  |

|  |                                                                                                                            |  |  |  |
|--|----------------------------------------------------------------------------------------------------------------------------|--|--|--|
|  | <ul style="list-style-type: none"> <li>▪ 1'b0: Session request failure</li> <li>▪ 1'b1: Session request success</li> </ul> |  |  |  |
|--|----------------------------------------------------------------------------------------------------------------------------|--|--|--|

### 33.5.3.2 Interrupt Register (GOTGINT)

◇ Offset: 004h

The application reads this register whenever there is an OTG interrupt and clears the bits in this register to clear the OTG interrupt. It is shown in Figure 33-2.

**Table 33-8 Interrupt Register: GOTGINT**

| Field  | Description                                                                                                                                                                                                                                                                                                                                       | Mode            | Reset | Access      |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-------|-------------|
| 31:20  | Reserved                                                                                                                                                                                                                                                                                                                                          | Host and Device |       | RO          |
| 20     | Multi-Valued input changed<br>This bit when set indicates that there is a change in the value of at least one ACA pin value. This bit is present only if OTG_BC_SUPPORT = 1, otherwise it is reserved.                                                                                                                                            |                 | 1'b0  | R_SS_W<br>C |
| 19     | Debounce Done (DbnceDone) The core sets this bit when the debounce is completed after the device connect. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively). | Host Only       | 1'b0  | R_SS_W<br>C |
| 18     | A-Device Timeout Change (ADevTOUTChg)<br>The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.                                                                                                                                                                                            | Host and Device | 1'b0  | R_SS_W<br>C |
| 17     | Host Negotiation Detected (HstNegDet)<br>The core sets this bit when it detects a host negotiation request on the USB.                                                                                                                                                                                                                            | Host and Device | 1'b0  | R_SS_W<br>C |
| 16: 10 | Reserved                                                                                                                                                                                                                                                                                                                                          | Host and Device |       | RO          |
| 9      | Host Negotiation Success Status Change (HstNegSucStsChng)<br>The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure.                                           | Host and Device | 1'b0  | R_SS_W<br>C |
| 8      | Session Request Success Status Change (SesReqSucStsChng)<br>The core sets this bit on the success or failure of a                                                                                                                                                                                                                                 | Host and Device | 1'b0  | R_SS_W<br>C |

|     |                                                                                                                                                                        |                 |      |         |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|---------|
|     | session request. The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure. |                 |      |         |
| 7:3 | Reserved                                                                                                                                                               | Host and Device |      | RO      |
| 2   | Session End Detected (SesEndDet)<br>The core sets this bit when the utmisrp_bvalid signal is deasserted.                                                               | Device only     | 1'b0 | R_SS_WC |
| 1:0 | Reserved                                                                                                                                                               | Host and Device |      | RO      |

### 33.5.3.3 AHB Configuration Register (GAHBCFG)

◇ Offset: 008h

This register can be used to configure the core after power-on or a change in mode. This register mainly contains AHB system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the AHB or the USB.

**Table 33-9 AHB Configuration Register: GAHBCFG**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Mode            | Reset | Access |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-------|--------|
| 31:23 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Host and Device |       | RO     |
| 23    | AHB Single Support (AHBSingle)<br>This bit when programmed supports Single transfers for the remaining data in a transfer when the DWC_otg core is operating in DMA mode. <ul style="list-style-type: none"> <li>▪ 1'b0: This is the default mode. When this bit is set to 1'b0, the remaining data in the transfer is sent using INCR burst size.</li> <li>▪ 1'b1: When set to 1'b1, the remaining data in a transfer is sent using Single burst size.</li> </ul> Note: if this feature is enabled, the AHB RETRY and SPLIT transfers still have INCR burst type. Enable this feature when the AHB Slave connected to the DWC_otg core does not support INCR burst (and when Split, and Retry transactions are not being used in the bus). | Host and Device | 1'b0  | R_W    |
| 22    | Notify All Dma Write Transactions (NotiAllDmaWrit)<br>This bit is programmed to enable the System DMA Done functionality for all the DMA write Transactions corresponding to the Channel/Endpoint. This bit is valid only when GAHBCFG.RemMemSupp is set to 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Host and Device | 1'b0  | R_W    |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                 |      |     |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|-----|
|      | <ul style="list-style-type: none"> <li>▪ GAHBCFG.NotiAllDmaWrit = 1 <ul style="list-style-type: none"> <li>- HS OTG core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint.</li> </ul> </li> <li>▪ GAHBCFG.NotiAllDmaWrit = 0 <ul style="list-style-type: none"> <li>- HS OTG core asserts int_dma_req signal only for the last transaction of DMA write transfer corresponding to a particular Channel/Endpoint. Similarly, the core waits for sys_dma_done signal only for that transaction of DMA write to complete the transfer of a particular Channel/Endpoint.</li> </ul> </li> </ul>                                                                                                                                                                                                                          |                 |      |     |
| 21   | <p>Remote Memory Support (RemMemSupp)</p> <p>This bit is programmed to enable the functionality to wait for the system DMA Done Signal for the DMA Write Transfers.</p> <ul style="list-style-type: none"> <li>▪ GAHBCFG.RemMemSupp=1 <ul style="list-style-type: none"> <li>- The int_dma_req output signal is asserted when HS OTG DMA starts write transfer to the external memory. When the core is done with the Transfers it asserts int_dma_done signal to flag the completion of DMA writes from HS OTG. The core then waits for sys_dma_done signal from the system to proceed further and complete the Data Transfer corresponding to a particular Channel/Endpoint.</li> </ul> </li> <li>▪ GAHBCFG.RemMemSupp=0 <ul style="list-style-type: none"> <li>- The int_dma_req and int_dma_done signals are not asserted and the core proceeds with the assertion of the XferComp interrupt as soon as the DMA write transfer is done at the HS OTG Core Boundary and it doesn't wait for the sys_dma_done signal to complete the DATA transfers.</li> </ul> </li> </ul> | Host and Device | 1'b0 | R_W |
| 20:9 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Host and Device |      | RO  |
| 8    | <p>Periodic Tx FIFO Empty Level (PTxFEmpLvl)</p> <p>Indicates when the Periodic Tx FIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.PTxFEmp) is</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Host only       | 1'b0 | R_W |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                 |      |     |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|-----|
|     | triggered. This bit is used only in Slave mode. <ul style="list-style-type: none"> <li>▪ 1'b0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty</li> <li>▪ 1'b1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |      |     |
| 7   | Non-Periodic TxFIFO Empty Level (NPTxFEmpLvl)<br>This bit is used only in Slave mode.<br>In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.NPTxFEmp) is triggered.<br>With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered.<br>Host mode and with Shared FIFO with device mode: <ul style="list-style-type: none"> <li>▪ 1'b0: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is half empty</li> <li>▪ 1'b1: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is completely empty</li> </ul> Dedicated FIFO in device mode: <ul style="list-style-type: none"> <li>▪ 1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty</li> <li>▪ 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty</li> </ul> | Host and Device | 1'b0 | R_W |
| 6   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Host and Device |      | RO  |
| 5   | DMA Enable (DMAEn) <ul style="list-style-type: none"> <li>▪ 1'b0: Core operates in Slave mode</li> <li>▪ 1'b1: Core operates in a DMA mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Host and Device | 1'b0 | R_W |
| 4:1 | Burst Length/Type (HBstLen)<br>This field is used in both External and Internal DMA modes. In External DMA mode, these bits appear on dma_burst[3:0] ports, which can be used by an external wrapper to interface the External DMA Controller interface to Synopsys DW_ahb_dmac or ARM PrimeCell.<br>External DMA Mode — defines the DMA burst length in terms of 32-bit words: <ul style="list-style-type: none"> <li>▪ 4'b0000: 1 word</li> <li>▪ 4'b0001: 4 words</li> <li>▪ 4'b0010: 8 words</li> <li>▪ 4'b0011: 16 words</li> <li>▪ 4'b0100: 32 words</li> <li>▪ 4'b0101: 64 words</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                         | Host and Device | 4'b0 | R_W |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                            |                 |      |     |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|-----|
|   | <ul style="list-style-type: none"> <li>▪ 4'b0110: 128 words</li> <li>▪ 4'b0111: 256 words</li> <li>▪ Others: Reserved</li> </ul> Internal DMA Mode — AHB Master burst type: <ul style="list-style-type: none"> <li>▪ 4'b0000 Single</li> <li>▪ 4'b0001 INCR</li> <li>▪ 4'b0011 INCR4</li> <li>▪ 4'b0101 INCR8</li> <li>▪ 4'b0111 INCR16</li> <li>▪ Others: Reserved</li> </ul> Slave Mode: Only single burst is supported. |                 |      |     |
| 0 | <b>Global Interrupt Mask (GblIntrMsk)</b><br>The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the core. <ul style="list-style-type: none"> <li>▪ 1'b0: Mask the interrupt assertion to the application.</li> <li>▪ 1'b1: Unmask the interrupt assertion to the application.</li> </ul>            | Host and Device | 1'b0 | R_W |

### 33.5.3.4 USB Configuration Register (GUSBCFG)

✧ Offset: 00Ch

This register can be used to configure the core after power-on or a changing to Host mode or Device mode.

It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the AHB or the USB. Do not make changes to this register after the initial programming.

**Table 33-10 USB Configuration Register: GUSBCFG**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                   | Mode            | Reset | Access |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-------|--------|
| 31    | <b>Corrupt Tx packet</b><br>This bit is for debug purposes only. Never set this bit to 1.                                                                                                                                                                                                                                                                                                                                                     | Host and Device | 1'b0  | WO     |
| 30    | <b>Force Device Mode (ForceDevMode)</b><br>Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. <ul style="list-style-type: none"> <li>▪ 1'b0: Normal Mode</li> <li>▪ 1'b1: Force Device Mode</li> </ul> After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 $\mu$ s is sufficient. | Host and Device | 1'b0  | R_W    |



|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                 |                  |        |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------------------|--------|
| 29 | <p>Force Host Mode (ForceHstMode)</p> <p>Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Normal Mode</li> <li>▪ 1'b1: Force Host Mode</li> </ul> <p>After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 <math>\mu</math>s is sufficient.</p> | Host and Device | 1'b0             | R_W    |
| 28 | <p>Tx End Delay (TxEndDelay)</p> <p>Writing a 1 to this bit enables the TxEndDelay timers in the core as per the section 4.1.5 on Opmode of the USB 2.0 Transceiver Macrocell Interface (UTMI) version 1.05.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Normal mode</li> <li>▪ 1'b1: Introduce Tx end delay timers</li> </ul>                                                                                                                        | Device Only     | 1'b0             | R_W    |
| 27 | <p>IC_USB TrafficPullRemove Control (IC_USBTrafCtl)</p> <p>When this bit is set, pullup/pulldown resistors are detached from the USB during traffic signaling, per section 6.3.4 of the IC_USB specification.</p> <p>This bit is invalid.</p>                                                                                                                                                                                                                | Device          | 1'h0             | R_W    |
| 26 | <p>IC_USB-Capable (IC_USBCap)</p> <p>The application uses this bit to control the DWC_otg core's IC_USB capabilities.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: IC_USB PHY Interface is not selected.</li> <li>▪ 1'b1: IC_USB PHY Interface is selected.</li> </ul> <p>This bit is set to 1'b0 and the bit is read only.</p>                                                                                                                        | Host and Device | Read description | RO/R_W |
| 25 | <p>ULPI Interface Protect Disable</p> <p>Controls circuitry built into the PHY for protecting the ULPI interface when the link tri-states STP and data. Any pull-ups or pull-downs employed by this feature can be disabled.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Enables the interface protect circuit</li> <li>▪ 1'b1: Disables the interface protect circuit</li> </ul> <p>This bit is reserved and read-only.</p>                          | Host only       | 1'b0             | R_W/RO |
| 24 | <p>Indicator Pass Through</p> <p>Controls whether the Complement Output is qualified with the Internal Vbus Valid comparator before being used in the Vbus State in the RX CMD. Please refer to the ULPI Specification for more detail.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Complement Output signal is qualified with the</li> </ul>                                                                                                         | Host Only       | 1'b0             | R_W/RO |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                |                 |      |        |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|--------|
|    | <p>Internal VbusValid comparator.</p> <ul style="list-style-type: none"> <li>1'b1: Complement Output signal is not qualified with the Internal VbusValid comparator.</li> </ul> <p>This bit is reserved and read-only.</p>                                                                                                                                                                                     |                 |      |        |
| 23 | <p>Indicator Complement</p> <p>Controls the PHY to invert the ExternalVbusIndicator input signal, generating the Complement Output. Please refer to the ULPI Specification for more detail</p> <ul style="list-style-type: none"> <li>1'b0: PHY does not invert ExternalVbusIndicator signal</li> <li>1'b1: PHY does invert ExternalVbusIndicator signal</li> </ul> <p>This bit is reserved and read-only.</p> | Host Only       | 1'b0 | R_W/RO |
| 22 | <p>TermSel DLine Pulsing Selection (TermSelDLPulse)</p> <p>This bit selects utmi_termselect to drive data line pulse during SRP.</p> <ul style="list-style-type: none"> <li>1'b0: Data line pulsing using utmi_txvalid (default).</li> <li>1'b1: Data line pulsing using utmi_termsel.</li> </ul>                                                                                                              | Device Only     | 1'b0 | R_W    |
| 21 | <p>ULPI External VBUS Indicator (ULPIExtVbusIndicator)</p> <p>This bit indicates to the ULPI PHY to use an external VBUS over-current indicator.</p> <ul style="list-style-type: none"> <li>1'b0: PHY uses internal VBUS valid comparator.</li> <li>1'b1: PHY uses external VBUS valid comparator.</li> </ul> <p>This bit is invalid.</p>                                                                      | Host Only       | 1'b0 | R_W    |
| 20 | <p>ULPI External VBUS Drive (ULPIExtVbusDrv)</p> <p>This bit selects between internal or external supply to drive 5V on VBUS, in ULPI PHY.</p> <ul style="list-style-type: none"> <li>1'b0: PHY drives VBUS using internal charge pump (default).</li> <li>1'b1: PHY drives VBUS using external supply.</li> </ul> <p>This bit is invalid.</p>                                                                 | Host Only       | 1'b0 | R_W    |
| 19 | <p>ULPI Clock SuspendM (ULPIClkSusM)</p> <p>This bit sets the ClockSuspendM bit in the Interface Control register on ULPI PHY.</p> <p>1'b0: PHY powers down internal clock during suspend.</p> <p>1'b1: PHY does not power down internal clock.</p> <p>This bit is invalid.</p>                                                                                                                                | Host and Device | 1'b0 | R_W    |
| 18 | <p>ULPI Auto Resume (ULPIAutoRes)</p> <p>This bit sets the AutoResume bit in the Interface Control register on the ULPIPHY.</p> <ul style="list-style-type: none"> <li>1'b0: PHY does not use AutoResume feature.</li> <li>1'b1: PHY uses AutoResume feature.</li> </ul> <p>This bit is invalid.</p>                                                                                                           | Host and Device | 1'b0 | R_W    |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |      |        |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|--------|
| 17    | <p>ULPI FS/LS Select (ULPIFsLs)</p> <p>The application uses this bit to select the FS/LS serial interface for the ULPI PHY. This bit is valid only when the FS serial transceiver is selected on the ULPI PHY.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: ULPI interface</li> <li>▪ 1'b1: ULPI FS/LS serial interface</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                       | Host and Device | 1'b0 | R_W    |
| 16    | <p>UTMIFS or I<sup>2</sup>C Interface Select (OtgI2CSel)</p> <p>The application uses this bit to select the I<sup>2</sup>C interface.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: UTMI USB 1.1 Full-Speed interface for OTG signals</li> <li>▪ 1'b1: I<sup>2</sup>C interface for OTG signals</li> </ul> <p>This bit is writable only if I<sup>2</sup>C and UTMIFS were specified for Enable I2C Interface</p>                                                                                                                                                                                                                                                                                                                          | Host and Device | 1'b0 | RO/R_W |
| 15    | <p>PHY Low-Power Clock Select (PhyLPwrClkSel)</p> <p>Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: 480-MHz Internal PLL clock</li> <li>▪ 1'b1: 48-MHz External Clock</li> </ul> <p>In 480 MHz mode, the UTMI interface operates at either 60 or 30-MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48 MHz in FS and LS modes.</p> <p>This bit drives the utmi_fsls_low_power core output signal, and is valid only for UTMI+ PHYs.</p>                                                                                            | Host and Device | 1'b0 | R_W    |
| 14    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Host and Device |      | RO     |
| 13:10 | <p>USB Turnaround Time (USBTrdTim) Sets the turnaround time in PHY clocks.</p> <p>Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM).</p> <p>This must be programmed to</p> <ul style="list-style-type: none"> <li>▪ 4'h5: When the MAC interface is 16-bit UTMI+.</li> <li>▪ 4'h9: When the MAC interface is 8-bit UTMI+.</li> </ul> <p><b>Note:</b> The values above are calculated for the minimum AHB frequency of 30 MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so if you need the AHB to run at less than 30 MHz, and if USB turnaround time is not critical, these bits can be programmed to a larger value.</p> | Device Only     | 4'h5 | R_W    |
| 9     | <p>HNP-Capable (HNPCap)</p> <p>The application uses this bit to control the DWC_otg</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Host and Device | 1'h0 | RO/R_W |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                 |      |        |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|--------|
|   | <p>core's HNP capabilities.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: HNP capability is not enabled.</li> <li>▪ 1'b1: HNP capability is enabled.</li> </ul> <p>If HNP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.</p>                                                                                                                                                                                                                          |                 |      |        |
| 8 | <p>SRP-Capable (SRPCap)</p> <p>The application uses this bit to control the DWC_otg core SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: SRP capability is not enabled.</li> <li>▪ 1'b1: SRP capability is enabled.</li> </ul> <p>If SRP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.</p> | Host and Device | 1'h0 | R_W    |
| 7 | <p>ULPI DDR Select (DDRSEL)</p> <p>The application uses this bit to select a Single Data Rate (SDR) or Double Data Rate (DDR) or ULPI interface.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Single Data Rate ULPI Interface, with 8-bit-wide data bus</li> <li>▪ 1'b1: Double Data Rate ULPI Interface, with 4-bit-wide data bus</li> </ul> <p>This bit is invalid.</p>                                                                                                                                                  | Host and Device | 1'h0 | R_W    |
| 6 | <p>USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select (PHYSEL)</p> <p>The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY</li> <li>▪ 1'b1: USB 1.1 full-speed serial transceiver</li> </ul> <p>This bit is always 0, with Read Only access.</p>                                                                                                              | Host and Device | 1'h0 | RO/R_W |
| 5 | <p>Full-Speed Serial Interface Select (FSIntf)</p> <p>The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: 6-pin unidirectional full-speed serial interface</li> <li>▪ 1'b1: 3-pin bidirectional full-speed serial interface</li> </ul> <p>This bit is always 0, with Read Only access.</p>                                                                                                      | Host and Device | 1'h0 | RO/R_W |
| 4 | <p>ULPI or UTMI+ Select (ULPI_UTMI_Sel)</p> <p>The application uses this bit to select either a UTMI+ interface or ULPI Interface.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: UTMI+ Interface</li> </ul>                                                                                                                                                                                                                                                                                                                 | Host and Device | 1'h0 | RO/R_W |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                 |              |        |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------|--------|
|     | <ul style="list-style-type: none"> <li>▪ 1'b1: ULPI Interface</li> </ul> <p>This bit takes effect only if GUSBCFG.PHYSel = 1'b0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                 |              |        |
| 3   | <p>PHY Interface (PHYIf)</p> <p>The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. When a ULPI PHY is chosen, this must be set to 8-bit mode.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: 8 bits</li> <li>▪ 1'b1: 16 bits</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Host and Device | Configurable | RO/R_W |
| 2:0 | <p>HS/FS Timeout Calibration (TOutCal)</p> <p>The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed interpacket timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the line state condition can vary from one PHY to another.</p> <p>The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are:</p> <p>High-speed operation:</p> <ul style="list-style-type: none"> <li>▪ One 30-MHz PHY clock = 16 bit times</li> <li>▪ One 60-MHz PHY clock = 8 bit times</li> </ul> <p>Full-speed operation:</p> <ul style="list-style-type: none"> <li>▪ One 30-MHz PHY clock = 0.4 bit times</li> <li>▪ One 60-MHz PHY clock = 0.2 bit times</li> <li>▪ One 48-MHz PHY clock = 0.25 bit times</li> </ul> <p>Using the HS as an example, if you set ToutCal to '001' you add one 30MHz PHY clock or 16 bit times. If you set ToutCal to '010' you add two 30MHz PHY clocks or 32 bit times, and so on. The 3 bits allow you to add up to 7 PHY clocks, and the number of bit times depend on the speed, and the PHY clock you are using.</p> | Host and Device | 3'h0         | R_W    |

### 33.5.3.5 Reset Register (GRSTCTL)

✧ Offset: 010h

The application uses this register to reset various hardware features inside the core.

Table 33-11 Reset Register: GRSTCTL

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Mode            | Reset | Access |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-------|--------|
| 31    | AHB Master Idle (AHBIdle)<br>Indicates that the AHB Master State Machine is in the IDLE condition.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Host and Device | 1'b1  | RO     |
| 30    | DMA Request Signal (DMAReq)<br>Indicates that the DMA request is in progress. Used for debug.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Host and Device | 1'b1  | RO     |
| 29:11 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Host and Device |       | RO     |
| 10:6  | <p>TxFIFO Number (TxFNum)<br/>This is the FIFO number that must be flushed using the TxFIFO Flush bit.<br/>This field must not be changed until the core clears the TxFIFO Flush bit.</p> <ul style="list-style-type: none"> <li>▪ 5'h0: <ul style="list-style-type: none"> <li>– Non-periodic TxFIFO flush in Host mode</li> <li>– Non-periodic TxFIFO flush in device mode when in shared FIFO operation</li> <li>– Tx FIFO 0 flush in device mode when in dedicated FIFO mode</li> </ul> </li> <li>▪ 5'h1: <ul style="list-style-type: none"> <li>– Periodic TxFIFO flush in Host mode</li> <li>– Periodic TxFIFO 1 flush in Device mode when in shared FIFO operation</li> <li>– TXFIFO 1 flush in device mode when in dedicated FIFO mode</li> </ul> </li> <li>▪ 5'h2: <ul style="list-style-type: none"> <li>– Periodic TxFIFO 2 flush in Device mode when in shared FIFO operation</li> <li>– TXFIFO 2 flush in device mode when in dedicated FIFO mode</li> </ul> </li> <li>...</li> <li>▪ 5'hF: <ul style="list-style-type: none"> <li>– Periodic TxFIFO 15 flush in Device mode when in shared FIFO operation</li> <li>– TXFIFO 15 flush in device mode when in dedicated FIFO mode</li> </ul> </li> <li>▪ 5'h10: <ul style="list-style-type: none"> <li>– Flush all the transmit FIFOs in device or host mode.</li> </ul> </li> </ul> | Host and Device | 5'b0  | R_W    |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |      |         |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|---------|
| 5 | <p><b>TxFIFO Flush (TxFFish)</b></p> <p>This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers:</p> <ul style="list-style-type: none"> <li>▪ Read—NAK Effective Interrupt ensures the core is not reading from the FIFO.</li> <li>▪ Write—GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO.</li> </ul> <p>Flushing is normally recommended when FIFOs are re-configured or when switching between Shared FIFO and Dedicated Transmit FIFO operation.</p> <p>FIFO flushing is also recommended during device endpoint disable.</p> <p>The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk.</p> | Host and Device | 1'b0 | R_WS_SC |
| 4 | <p><b>RxFIFO Flush (RxFFish)</b></p> <p>The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction.</p> <p>The application must only write to this bit after checking that the core is neither reading from the RxFIFO nor writing to the RxFIFO.</p> <p>The application must wait until the bit is cleared before performing any other operations. This bit requires 8 clocks (slowest of PHY or AHB clock) to clear.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Host and Device | 1'b0 | R_WS_SC |
| 3 | <p><b>IN Token Sequence Learning Queue Flush (INTknQFish)</b></p> <p>This bit is valid only if OTG_EN_DED_TX_FIFO = 0.</p> <p>The application writes this bit to flush the IN Token Sequence Learning Queue.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Device Only     | 1'b0 | R_WS_SC |
| 2 | <p><b>Host Frame Counter Reset (FrmCntRst)</b></p> <p>The application writes this bit to reset the (micro)frame number counter inside the core. When the (micro)frame counter is reset, the subsequent SOF sent out by the core has a (micro)frame number of 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Host Only       | 1'b0 | R_WS_SC |
| 1 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Host and Device |      | RO      |
| 0 | <p><b>Core Soft Reset (CSftRst)</b></p> <p>Resets the hclk and phy_clock domains as follows:</p> <ul style="list-style-type: none"> <li>▪ Clears the interrupts and all the CSR registers except the following register bits:</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Host and Device | 1'b0 | R_WS_SC |

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |  |  |  |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|
|  | <ul style="list-style-type: none"> <li>- PCGCCTL.RstPdownModule</li> <li>- PCGCCTL.GateHclk</li> <li>- PCGCCTL.PwrClmp</li> <li>- GUSBCFG.DDRSel</li> <li>- GUSBCFG.PHYSel</li> <li>- GUSBCFG.FSIntf</li> <li>- GUSBCFG.ULPI_UTMI_Sel</li> <li>- GUSBCFG.PHYIf</li> <li>- HCFG.FSLSPclkSel</li> <li>- DCFG.DevSpd</li> <li>- DCTL.SftDiscon</li> <li>- GGPIO</li> <li>- GUSBCFG.TxEndDelay</li> <li>- GUSBCFG.TermSelDLPulse</li> <li>- GUSBCFG.ULPIClkSusM</li> <li>- GUSBCFG.ULPIAutoRes</li> <li>- GUSBCFG.ULPIFsLs</li> <li>- GPWRDN</li> <li>- GADPCTL</li> </ul> <ul style="list-style-type: none"> <li>▪ All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed.</li> <li>▪ Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately.</li> <li>▪ When Hibernation or ADP feature is enabled, the PMU module is not reset by the Core Soft Reset.</li> </ul> <p>The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). Software must also must check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation.</p> <p>Typically software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain.</p> |  |  |  |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|



|  |                                                                                    |  |  |  |
|--|------------------------------------------------------------------------------------|--|--|--|
|  | Once a new clock is selected, the PHY domain has to be reset for proper operation. |  |  |  |
|--|------------------------------------------------------------------------------------|--|--|--|

### 33.5.3.6 Interrupt Register (GINTSTS)

◇ Offset: 014h

This register interrupts the application for system-level events in the current mode (Device mode or Host mode). It is shown in Figure 33-2.

Some of the bits in this register are valid only in Host mode, while others are valid in Device mode only. This register also indicates the current mode. To clear the interrupt status bits of type R\_SS\_WC, the application must write 1'b1 into the bit.

The FIFO status interrupts are read only; once software reads from or writes to the FIFO while servicing these interrupts, FIFO interrupt conditions are cleared automatically.

The application must clear the GINTSTS register at initialization before unmasking the interrupt bit to avoid any interrupts generated prior to initialization.

**Table 33-12 Interrupt Register: GINTSTS**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Mode                  | Reset | Access      |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|-------|-------------|
| 31    | Resume/Remote Wakeup Detected Interrupt (WkUpInt)<br>Wakeup Interrupt during Suspend(L2) or LPM(L1) state. <ul style="list-style-type: none"> <li>▪ During Suspend(L2):                             <ul style="list-style-type: none"> <li>– Device Mode - This interrupt is asserted only when Host Initiated Resume is detected on USB.</li> <li>– Host Mode - This interrupt is asserted only when Device Initiated Remote Wakeup is detected on USB.</li> </ul> </li> <li>▪ During LPM(L1):                             <ul style="list-style-type: none"> <li>– Device Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.</li> <li>– Host Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.</li> </ul> </li> </ul> | Host<br>and<br>Device | 1'b0  | R_SS_W<br>C |
| 30    | Session Request/New Session Detected Interrupt (SessReqInt)<br>In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when a session request is detected                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Host<br>and<br>Device | 1'b0  | R_SS_W<br>C |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                 |      |             |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|-------------|
|    | <p>from the device.</p> <p>In Device mode, this interrupt is asserted when the utmisrp_bvalid signal goes high.</p> <p>For more information on how to use this interrupt, see "FIFO RAM Allocation" in the Programming Guide.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |      |             |
| 29 | <p>Disconnect Detected Interrupt (DisconnInt)</p> <p>This interrupt is asserted when a device disconnect is detected.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Host Only       | 1'b0 | R_SS_W<br>C |
| 28 | <p>Connector ID Status Change (ConIDStsChng)</p> <p>This interrupt is asserted when there is a change in connector ID status.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Host and Device | 1'b0 | R_SS_W<br>C |
| 27 | <p>LPM Transaction Received Interrupt (LPM_Int)</p> <ul style="list-style-type: none"> <li>▪ Device Mode - This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERRORed response.</li> <li>▪ Host Mode - This interrupt is asserted when the device responds to an LPM transaction with a non-ERRORed response or when the host core has completed LPM transactions for the programmed number of times (GLPMCFCFG.RetryCnt).</li> </ul> <p>This field is valid only if the Core LPM Configuration register's LPM- Capable (LPMCap) field is set to 1 and the User HW Config3 register's OTG_ENABLE_LPM bit is set to 1.</p> | Host and Device | 1'b0 | R_SS_W<br>C |
| 26 | <p>Periodic Tx FIFO Empty (PTxFEmp)</p> <p>This interrupt is asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic Tx FIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.PTxFEmpLvl).</p>                                                                                                                                                                                                                                                                            | Host Only       | 1'b1 | RO          |
| 25 | <p>Host Channels Interrupt (HChInt)</p> <p>The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.</p>                                                                                                                  | Host Only       | 1'b0 | RO          |
| 24 | <p>Host Port Interrupt (PrtInt)</p> <p>The core sets this bit to indicate a change in port status of</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Host Only       | 1'b0 | RO          |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |      |         |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|------|---------|
|    | <p>one of the DWC_otg core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |             |      |         |
| 23 | <p><b>Reset Detected Interrupt (ResetDet)</b><br/>The core sets this status bit in device mode when reset is detected on the USB in L2 Suspend state.<br/>This bit is valid only when the core is Device Mode and is operating in Partial Power-Down, or Clock Gating modes of Suspend.<br/>This bit is not valid when device is in Hibernation mode of Suspend.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Device Only | 1'b0 | R_SS_WC |
| 22 | <p><b>Data Fetch Suspended (FetSusp)</b><br/>This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm.<br/>For example, after detecting an endpoint mismatch, the application:</p> <ul style="list-style-type: none"> <li>▪ Sets a global non-periodic IN NAK handshake</li> <li>▪ Disables In endpoints</li> <li>▪ Flushes the FIFO</li> <li>▪ Determines the token sequence from the IN Token Sequence Learning Queue</li> <li>▪ Re-enables the endpoints</li> <li>▪ Clears the global non-periodic IN NAK handshake</li> </ul> <p>If the global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received: the core generates an "IN token received when FIFO empty" interrupt. The OTG then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application can mask the "IN token received when FIFO empty" interrupt when clearing a global IN NAK handshake.</p> | Device Only | 1'b0 | R_SS_WC |
| 21 | <p><b>Incomplete Periodic Transfer (incomplIP)</b><br/>In Host mode, the core sets this interrupt bit when there are</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Device only |      |         |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |      |             |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|------|-------------|
|    | <p>incomplete periodic transactions still pending which are scheduled for the current microframe.</p> <p>Incomplete Isochronous OUT Transfer (incomplISOOUT)</p> <p>The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.</p>                                                                                                                                                            | Device only |      |             |
| 20 | <p>Incomplete Isochronous IN Transfer (incomplISOIN)</p> <p>The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.</p> <p>Note: This interrupt is not asserted in Scatter/Gather DMA mode.</p>                                                                                                                                                                                                              | Device only | 1'b0 | R_SS_W<br>C |
| 19 | <p>OUT Endpoints Interrupt (OEPInt)</p> <p>The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-<i>n</i> Interrupt (DOEPINT<sub>n</sub>) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINT<sub>n</sub> register to clear this bit.</p> | Device only | 1'b0 | RO          |
| 18 | <p>IN Endpoints Interrupt (IEPInt)</p> <p>The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding Device IN Endpoint-<i>n</i> Interrupt (DIEPINT<sub>n</sub>) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINT<sub>n</sub> register to clear this bit.</p>     | Device only | 1'b0 | RO          |
| 17 | <p>Endpoint Mismatch Interrupt (EPMis)</p> <p><b>Note:</b> This interrupt is valid only in shared FIFO operation. Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and</p>                                                                                                                                                                                                                                                                                                                             | Device only | 1'b0 | R_SS_W<br>C |

|    |                                                                                                                                                                                                                                                                                                                                                     |                      |      |             |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|------|-------------|
|    | the IN endpoint mismatch count programmed by the application has expired.                                                                                                                                                                                                                                                                           |                      |      |             |
| 16 | <p>Restore Done Interrupt (RstrDoneInt)</p> <p>The core sets this bit to indicate that the restore command after Hibernation was completed by the core.</p> <p>The core continues from Suspended state into the mode dictated by PCGCCTL.RestoreMode field.</p> <p>This bit is valid only when Hibernation feature is enabled (OTG_EN_PWROPT=2)</p> | Host and Device mode | 1'b0 | R_SS_W<br>C |
| 15 | <p>End of Periodic Frame Interrupt (EOPF)</p> <p>Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.</p>                                                                                                                      | Device only          | 1'b0 | R_SS_W<br>C |
| 14 | <p>Isochronous OUT Packet Dropped Interrupt (ISOOutDrop)</p> <p>The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.</p>                                                             | Device only          | 1'b0 | R_SS_W<br>C |
| 13 | <p>Enumeration Done (EnumDone)</p> <p>The core sets this bit to indicate that speed enumeration is complete. The application must read the <a href="#">Device Status (DSTS)</a> register to obtain the enumerated speed.</p>                                                                                                                        | Device only          | 1'b0 | R_SS_W<br>C |
| 12 | <p>USB Reset (USBRst)</p> <p>The core sets this bit to indicate that a reset is detected on the USB.</p>                                                                                                                                                                                                                                            | Device only          | 1'b0 | R_SS_W<br>C |
| 11 | <p>USB Suspend (USBSusp)</p> <p>The core sets this bit to indicate that a suspend was detected on the USB.</p> <p>The core enters the Suspended state when there is no activity on the utmi_linestate signal for an extended period of time.</p>                                                                                                    | Device only          | 1'b0 | R_SS_W<br>C |
| 10 | <p>Early Suspend (ErlySusp)</p> <p>The core sets this bit to indicate that an Idle state has been detected on the USB for 3 ms.</p>                                                                                                                                                                                                                 | Device only          | 1'b0 | R_SS_W<br>C |
| 9  | <p>I<sup>2</sup>C Interrupt (I2CINT)</p> <p>The core sets this interrupt when I<sup>2</sup>C access is completed on the I<sup>2</sup>C interface. This field is used only if the I<sup>2</sup>C interface was enabled. Otherwise, reads return 0.</p>                                                                                               | Host and Device      | 1'b0 | R_SS_W<br>C |
| 8  | <p>ULPI Carkit Interrupt (ULPICKINT)</p> <p>This field is used only if the Carkit interface was enabled. Otherwise, reads return 0.</p>                                                                                                                                                                                                             | Host and Device      | 1'b0 | R_SS_W<br>C |
| 7  | <p>Global OUT NAK Effective (GOUTNakEff)</p>                                                                                                                                                                                                                                                                                                        | Device               | 1'b0 | RO          |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                 |      |             |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|-------------|
|   | Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | only            |      |             |
| 6 | Global IN Non-Periodic NAK Effective (GINNakEff)<br>Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak), set by the application, has taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Non-periodic IN NAK bit in the Device Control register (DCTL.CGNPInNak).<br>This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.                                                                                                                                                                                                      | Device only     | 1'b0 | RO          |
| 5 | Non-Periodic Tx FIFO Empty (NPTxFEmp)<br>This interrupt is valid only when OTG_EN_DED_TX_FIFO = 0.<br>This interrupt is asserted when the Non-periodic Tx FIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic Tx FIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl).                                                                                                                                                                                                                                                                                                        | Host and Device | 1'b1 | RO          |
| 4 | RxFIFO Non-Empty (RxFLvl)<br>Indicates that there is at least one packet pending to be read from the RxFIFO.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Host and Device | 1'b0 | RO          |
| 3 | Start of (micro)Frame (Sof)<br>In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt.<br>In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro)frame number. This interrupt is seen only when the core is operating at either HS or FS.<br><b>Note:</b> This register may return 1'b1 if read immediately after power on reset. If the register bit reads 1'b1 immediately after power on reset it does not indicate that an SOF has been sent (in case of host mode) or SOF has been received | Host and Device | 1'b0 | R_SS_W<br>C |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |      |             |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|-------------|
|   | (in case of device mode). The read value of this interrupt is valid only after a valid connection between host and device is established. If the bit is set after power on reset the application can clear the bit.                                                                                                                                                                                                                                                 |                 |      |             |
| 2 | <b>OTG Interrupt (OTGInt)</b><br>The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit.                                                                                                                                          | Host and Device | 1'b0 | RO          |
| 1 | <b>Mode Mismatch Interrupt (ModeMis)</b><br>The core sets this bit when the application is trying to access: <ul style="list-style-type: none"> <li>▪ A Host mode register, when the core is operating in Device mode</li> <li>▪ A Device mode register, when the core is operating in Host mode</li> </ul> The register access is completed on the AHB with an OKAY response, but is ignored by the core internally and does not affect the operation of the core. | Host and Device | 1'b0 | R_SS_W<br>C |
| 0 | <b>Current Mode of Operation (CurMod)</b><br>Indicates the current mode. <ul style="list-style-type: none"> <li>▪ 1'b0: Device mode</li> <li>▪ 1'b1: Host mode</li> </ul>                                                                                                                                                                                                                                                                                           | Host and Device | 1'b0 | RO          |

### 33.5.3.7 Interrupt Mask Register (GINTMSK)

✧ Offset: 018h

This register works with the "[Interrupt Register \(GINTSTS\)](#)" to interrupt the application. When an interrupt bit is masked, the interrupt associated with that bit is not generated. However, the GINTSTS register bit corresponding to that interrupt is still set.

✧ Mask interrupt: 1'b0

✧ Unmask interrupt: 1'b1

**Table 33-13 Interrupt Mask Register: GINTMSK**

| Field | Description                                                         | Mode            | Reset | Access |
|-------|---------------------------------------------------------------------|-----------------|-------|--------|
| 31    | Resume/Remote Wakeup Detected Interrupt Mask (WkUpIntMsk)           | Host and Device | 1'b0  | R_W    |
| 30    | Session Request/New Session Detected Interrupt Mask (SessReqIntMsk) | Host and Device | 1'b0  | R_W    |
| 29    | Disconnect Detected Interrupt Mask (DisconnIntMsk)                  | Host            | 1'b0  | R_W    |

|    |                                                                                                                                                               |                              |      |     |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|------|-----|
|    |                                                                                                                                                               | Only                         |      |     |
| 28 | Connector ID Status Change Mask (ConIDStsChngMsk)                                                                                                             | Host and Device              | 1'b0 | R_W |
| 27 | LPM Transaction Received Interrupt Mask (LPM_IntMsk)                                                                                                          |                              | 1'b0 |     |
| 26 | Periodic TxFIFO Empty Mask (PTxFEmpMsk)                                                                                                                       | Host Only                    | 1'b1 | R_W |
| 25 | Host Channels Interrupt Mask (HChIntMsk)                                                                                                                      | Host Only                    | 1'b0 | R_W |
| 24 | Host Port Interrupt Mask (PrtIntMsk)                                                                                                                          | Host Only                    | 1'b0 | R_W |
| 23 | Reset Detected Interrupt Mask (ResetDetMsk)                                                                                                                   | Device Only                  | 1'b0 | R_W |
| 22 | Data Fetch Suspended Mask (FetSuspMsk)                                                                                                                        | Device Only                  | 1'b0 | R_W |
| 21 | Incomplete Periodic Transfer Mask (incomplPMsk)<br><br>Incomplete Isochronous OUT Transfer Mask (incomplSOOUTMsk)                                             | Host only<br><br>Device only | 1'b0 | R_W |
| 20 | Incomplete Isochronous IN Transfer Mask (incomplSOINMsk)<br><br>This bit is enabled only when device periodic endpoints are enabled in Dedicated TxFIFO mode. | Device only                  | 1'b0 | R_W |
| 19 | OUT Endpoints Interrupt Mask (OEPIntMsk)                                                                                                                      | Device only                  | 1'b0 | R_W |
| 18 | IN Endpoints Interrupt Mask (IEPIntMsk)                                                                                                                       | Device only                  | 1'b0 | R_W |
| 17 | Endpoint Mismatch Interrupt Mask (EPMisMsk)                                                                                                                   | Device only                  | 1'b0 | R_W |
| 16 | Restore Done Interrupt Mask (RstrDoneIntMsk)<br><br>This field is valid only when Hibernation feature is enabled (OTG_EN_PWROPT=2).                           | Host and Device mode         | 1'b0 | R_W |
| 15 | End of Periodic Frame Interrupt Mask (EOPFMsk)                                                                                                                | Device only                  | 1'b0 | R_W |
| 14 | Isochronous OUT Packet Dropped Interrupt Mask (ISOOutDropMsk)                                                                                                 | Device only                  | 1'b0 | R_W |
| 13 | Enumeration Done Mask (EnumDoneMsk)                                                                                                                           | Device only                  | 1'b0 | R_W |
| 12 | USB Reset Mask (USBRstMsk)                                                                                                                                    | Device only                  | 1'b0 | R_W |
| 11 | USB Suspend Mask (USBSuspMsk)                                                                                                                                 | Device only                  | 1'b0 | R_W |
| 10 | Early Suspend Mask (ErlySuspMsk)                                                                                                                              | Device                       | 1'b0 | R_W |



|   |                                                                                                   |                 |      |     |
|---|---------------------------------------------------------------------------------------------------|-----------------|------|-----|
|   |                                                                                                   | only            |      |     |
| 9 | I <sup>2</sup> C Interrupt Mask (I2CIntMsk)                                                       | Host and Device | 1'b0 | R_W |
| 8 | ULPI Carkit Interrupt Mask (ULPICKINTMsk)<br>I <sup>2</sup> C Carkit Interrupt Mask (I2CCKINTMsk) | Host and Device | 1'b0 | R_W |
| 7 | Global OUT NAK Effective Mask (GOUTNakEffMsk)                                                     | Device only     | 1'b0 | R_W |
| 6 | Global Non-periodic IN NAK Effective Mask (GINNakEffMsk)                                          | Device only     | 1'b0 | R_W |
| 5 | Non-periodic Tx FIFO Empty Mask (NPTxFEmpMsk)                                                     | Host and Device | 1'b1 | R_W |
| 4 | Receive FIFO Non-Empty Mask (RxFLvlMsk)                                                           | Host and Device | 1'b0 | R_W |
| 3 | Start of (micro)Frame Mask (SofMsk)                                                               | Host and Device | 1'b0 | R_W |
| 2 | OTG Interrupt Mask (OTGIntMsk)                                                                    | Host and Device | 1'b0 | R_W |
| 1 | Mode Mismatch Interrupt Mask (ModeMisMsk)                                                         | Host and Device | 1'b0 | R_W |
| 0 | Reserved                                                                                          | Host and Device |      | RO  |

### 33.5.3.8 Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)

- ✧ Offset for Read: 01Ch
- ✧ Offset for Pop: 020h

A read to the Receive Status Debug Read register returns the contents of the top of the Receive FIFO. A read to the Receive Status Read and Pop register additionally pops the top data entry out of the Rx FIFO.

The receive status contents must be interpreted differently in Host and Device modes. The core ignores the receive status pop/read when the receive FIFO is empty and returns a value of 32'h0000\_0000. The application must only pop the Receive Status FIFO when the Receive FIFO Non-Empty bit of the Core Interrupt register (GINTSTS.RxFLvl) is asserted.

**Note:**

- Use of these fields vary based on whether the HS OTG core is functioning as a host or a device.
- Do not read this register's reset value before configuring the core because the read value is "X" in the simulation.

Following table shows Host mode.

**Table 33-14 Host Mode Receive Status Debug Read/Status Read and Pop Registers:  
GRXSTSR/GRXSTSP**

| Field | Description                                                                                                                                                                                                                                                                                                                        | Reset | Acces |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:21 | Reserved                                                                                                                                                                                                                                                                                                                           |       | RO    |
| 20:17 | Packet Status (PktSts) <ul style="list-style-type: none"> <li>▪ 4'b0010: IN data packet received</li> <li>▪ 4'b0011: IN transfer completed (triggers an interrupt)</li> <li>▪ 4'b0101: Data toggle error (triggers an interrupt)</li> <li>▪ 4'b0111: Channel halted (triggers an interrupt)</li> <li>▪ Others: Reserved</li> </ul> | 4'b0  | RO    |
| 16:15 | Data PID (DPID)<br>Indicates the Data PID of the received packet                                                                                                                                                                                                                                                                   | 2'b0  | RO    |
| 14:4  | Byte Count (BCnt)<br>Indicates the byte count of the received IN data packet.                                                                                                                                                                                                                                                      | 11'b0 | RO    |
| 3:0   | Channel Number (ChNum)<br>Indicates the channel number to which the current received packet belongs.                                                                                                                                                                                                                               | 4'h0  | RO    |

Following table shows Device mode

**Table 33-15 Device Mode Receive Status Debug Read/Status Read and Pop Registers:  
GRXSTSR/GRXSTSP**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset | Acces |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:25 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                     |       | RO    |
| 24:21 | Frame Number (FN)<br>This is the least significant 4 bits of the (micro)frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.                                                                                                                                                                                                                                  | 4'b0  | RO    |
| 20:17 | Packet Status (PktSts)<br>Indicates the status of the received packet <ul style="list-style-type: none"> <li>▪ 4'b0001: Global OUT NAK (triggers an interrupt)</li> <li>▪ 4'b0010: OUT data packet received</li> <li>▪ 4'b0011: OUT transfer completed (triggers an interrupt)</li> <li>▪ 4'b0100: SETUP transaction completed (triggers an interrupt)</li> <li>▪ 4'b0110: SETUP data packet received</li> <li>▪ Others: Reserved</li> </ul> | 4'b0  | RO    |
| 16:15 | Data PID (DPID)<br>Indicates the Data PID of the received OUT data packet <ul style="list-style-type: none"> <li>▪ 2'b00: DATA0</li> <li>▪ 2'b10: DATA1</li> <li>▪ 2'b01: DATA2</li> <li>▪ 2'b11: MDATA</li> </ul>                                                                                                                                                                                                                           | 2'b0  | RO    |

|      |                                                                                                        |       |    |
|------|--------------------------------------------------------------------------------------------------------|-------|----|
| 14:4 | Byte Count (BCnt)<br>Indicates the byte count of the received data packet.                             | 11'h0 | RO |
| 3:0  | Endpoint Number (EPNum)<br>Indicates the endpoint number to which the current received packet belongs. | 4'b0  | RO |

### 33.5.3.9 Receive FIFO Size Register (GRXFSIZ)

◇ Offset: 024h

The application can program the RAM size that must be allocated to the RxFIFO.

**Table 33-16 Receive FIFO Size Register: GRXFSIZ**

| Field | Description                                                                                                                                                                | Reset    | Acces |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------|
| 31:16 | Reserved                                                                                                                                                                   |          | RO    |
| 15:0  | RxFIFO Depth (RxFDep)<br>This value is in terms of 32-bit words. <ul style="list-style-type: none"> <li>▪ Minimum value is 16</li> <li>▪ Maximum value is 3648.</li> </ul> | 16'd3648 | R_W   |

### 33.5.3.10 Non-Periodic Transmit FIFO Size Register (GNPTXFSIZ)

◇ Offset: 028h

The application can program the RAM size and the memory start address for the Non-periodic TxFIFO.

**Note:** The fields of this register change, depending on host or device mode.

**Table 33-17 Non-Periodic Transmit FIFO Size Register: GNPTXFSIZ (Host Mode and Device Shared FIFO Mode)**

| Field | Description                                                                                                                                                                                                                                                                                                                                               | Reset        | Acces  |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31:16 | Non-periodic TxFIFO Depth (NPTxFDep)<br>For host mode, this field is always valid.<br>For Device mode, this field is invalid.<br>This value is in terms of 32-bit words. <ul style="list-style-type: none"> <li>▪ Minimum value is 16</li> <li>▪ Maximum value is 3648</li> </ul>                                                                         | Configurable | RO/R_W |
| 15:0  | Non-periodic Transmit RAM Start Address (NPTxFStAddr)<br>For host mode, this field is always valid.<br>This field contains the memory start address for Non-periodic Transmit FIFO RAM. <ul style="list-style-type: none"> <li>▪ The application can write a new value in this field. Programmed values must not exceed the power-on value set</li> </ul> | Configurable | R_W    |

**Table 33-18 Non-Periodic Transmit FIFO Size Register: GNPTXFSIZ (Device Dedicated FIFO Mode)**

| Field | Description                                                                                                                                                                                                                                           | Reset        | Acces  |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31:16 | IN Endpoint Tx FIFO 0 Depth (INEPTxF0Dep) This field is valid only for Device mode .<br>This value is in terms of 32-bit words. <ul style="list-style-type: none"> <li>▪ Minimum value is 16</li> <li>▪ Maximum value is 3648</li> </ul>              | Configurable | RO/R_W |
| 15:0  | IN Endpoint FIFO0 Transmit RAM Start Address (INEPTxF0StAddr)<br>For Device mode, this field is valid.<br>This field contains the memory start address for IN Endpoint Transmit FIFO# 0.<br>Programmed values must not exceed the power-on value set. | Configurable | RO/R_W |

### 33.5.3.11 Non-Periodic Transmit FIFO/Queue Status Register (GNPTXSTS)

◇ Offset: 02Ch

In Device mode, this register is valid only in Shared FIFO operation.

This read-only register contains the free space information for the Non-periodic Tx FIFO and the Non-periodic Transmit Request Queue.

**Table 33-19 Non-Periodic Transmit FIFO/Queue Status Register: GNPTXSTS**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset        | Acces |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-------|
| 31:24 | Top of the Non-periodic Transmit Request Queue (NPTxQTop)<br>Entry in the Non-periodic Tx Request Queue that is currently being processed by the MAC. <ul style="list-style-type: none"> <li>▪ Bits [30:27]: Channel/endpoint number</li> <li>▪ Bits [26:25]: <ul style="list-style-type: none"> <li>– 2'b00: IN/OUT token</li> <li>– 2'b01: Zero-length transmit packet (device IN/host OUT)</li> <li>– 2'b10: PING/CSPLIT token</li> <li>– 2'b11: Channel halt command</li> </ul> </li> <li>▪ Bit [24]: Terminate (last entry for selected channel/endpoint)</li> </ul> | 7'b0         | RO    |
| 23:16 | Non-periodic Transmit Request Queue Space Available (NPTxQSpcAvail) Indicates the amount of free space available in the Non-periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. Device mode has only IN requests. <ul style="list-style-type: none"> <li>▪ 8'h0: Non-periodic Transmit Request Queue is full</li> <li>▪ 8'h1: 1 location available</li> <li>▪ 8'h2: 2 locations available</li> </ul>                                                                                                                                 | Configurable | RO    |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |              |    |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|----|
|      | <ul style="list-style-type: none"> <li>▪ n : n locations available (<math>0 \leq n \leq 8</math>)</li> <li>▪ Others: Reserved</li> </ul>                                                                                                                                                                                                                                                                                                                                                                            |              |    |
| 15:0 | <p>Non-periodic TxFIFO Space Avail (NPTxFSpAvail)</p> <p>Indicates the amount of free space available in the Non-periodic TxFIFO.</p> <p>Values are in terms of 32-bit words.</p> <ul style="list-style-type: none"> <li>▪ 16'h0: Non-periodic TxFIFO is full</li> <li>▪ 16'h1: 1 word available</li> <li>▪ 16'h2: 2 words available</li> <li>▪ 16'h<i>n</i>: <i>n</i> words available (where <math>0 \leq n \leq 32,768</math>)</li> <li>▪ 16'h8000: 32,768 words available</li> <li>▪ Others: Reserved</li> </ul> | Configurable | RO |

### 33.5.3.12 User HW Config1 Register (GHWCFG1)

✧ Offset: 044

This register contains the logical endpoint direction(s) selected.

**Table 33-20 User HW Config1 Register: GHWCFG1**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset        | Access |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31:0  | <p>Endpoint Direction (epdir)</p> <p>This 32-bit field uses two bits per endpoint to determine the endpoint direction.</p> <p>Endpoint</p> <ul style="list-style-type: none"> <li>▪ Bits [31:30]: Endpoint 15 direction</li> <li>▪ Bits [29:28]: Endpoint 14 direction</li> <li>...</li> <li>▪ Bits [3:2]: Endpoint 1 direction</li> <li>▪ Bits [1:0]: Endpoint 0 direction (always BIDIR)</li> </ul> <p>Direction</p> <ul style="list-style-type: none"> <li>▪ 2'b00: BIDIR (IN and OUT) endpoint</li> <li>▪ 2'b01: IN endpoint</li> <li>▪ 2'b10: OUT endpoint</li> <li>▪ 2'b11: Reserved</li> </ul> <p>This field is configured using "Name: OTG_EP_DIR_1(n)".</p> | Configurable | RO     |

### 33.5.3.13 User HW Config2 Register (GHWCFG2)

✧ Offset: 048h

This register contains configuration options selected.

Table 33-21 User HW Config2 Register: GHWCFG2

| Field | Description                                                                                                                                                                                                                                                | Reset        | Access |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31    | OTG_ENABLE_IC_USB<br>IC_USB mode specified for mode of operation (parameter OTG_ENABLE_IC_USB) in coreConsultant.<br>To choose IC_USB_MODE, both OTG_FSPHY_INTERFACE and OTG_ENABLE_IC_USB must be 1.                                                      | Configurable | RO     |
| 30:26 | Device Mode IN Token Sequence Learning Queue Depth (TknQDepth) Range: 0-30<br>This field is configured using "Name: OTG_TOKEN_QUEUE_DEPTH".                                                                                                                | Configurable | RO     |
| 25:24 | Host Mode Periodic Request Queue Depth (PTxQDepth) <ul style="list-style-type: none"> <li>▪ 2'b00: 2</li> <li>▪ 2'b01: 4</li> <li>▪ 2'b10: 8</li> <li>▪ 2'b11: 16</li> </ul> This field is configured using parameter "Name: OTG_PERIO_TX_QUEUE_DEPTH"     | Configurable | RO     |
| 23:22 | Non-periodic Request Queue Depth (NPTxQDepth) <ul style="list-style-type: none"> <li>▪ 2'b00: 2</li> <li>▪ 2'b01: 4</li> <li>▪ 2'b10: 8</li> <li>▪ Others: Reserved</li> </ul> This field is configured using parameter "Name: OTG_NPERIO_TX_QUEUE_DEPTH". | Configurable | RO     |
| 21    | Reserved                                                                                                                                                                                                                                                   |              | RO     |
| 20    | Multi Processor Interrupt Enabled (MultiProclntrpt) <ul style="list-style-type: none"> <li>▪ 1'b0: No</li> <li>▪ 1'b1: Yes</li> </ul> This field is configured using parameter "Name: OTG_MULTI_PROC_INTRPT".                                              | Configurable | RO     |
| 19    | Dynamic FIFO Sizing Enabled (DynFifoSizing) <ul style="list-style-type: none"> <li>▪ 1'b0: No</li> <li>▪ 1'b1: Yes</li> </ul> This field is configured using parameter "Name: OTG_DFIFO_DYNAMIC".                                                          | Configurable | RO     |
| 18    | Periodic OUT Channels Supported in Host Mode (PerioSupport) <ul style="list-style-type: none"> <li>▪ 1'b0: No</li> <li>▪ 1'b1: Yes</li> </ul> This field is configured using parameter "Name: OTG_EN_PERIO_HOST".                                          | Configurable | RO     |
| 17:14 | Number of Host Channels (NumHstChnl)                                                                                                                                                                                                                       | Configurable | RO     |

|       |                                                                                                                                                                                                                                                                                                                                                                                               |              |    |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|----|
|       | <p>Indicates the number of host channels supported by the core in Host mode. The range of this field is 0-15: 0 specifies 1 channel, 15 specifies 16 channels.</p> <p>This field is configured using parameter "Name: <a href="#">OTG_NUM_HOST_CHAN</a>".</p>                                                                                                                                 |              |    |
| 13:10 | <p>Number of Device Endpoints (NumDevEps)</p> <p>Indicates the number of device endpoints supported by the core in Device mode in addition to control endpoint 0. The range of this field is 1-15.</p> <p>This field is configured using parameter "Name: <a href="#">OTG_NUM_EPS</a>".</p>                                                                                                   | Configurable | RO |
| 9:8   | <p>Full-Speed PHY Interface Type (FSPhyType)</p> <ul style="list-style-type: none"> <li>▪ 2'b00: Full-speed interface not supported</li> <li>▪ 2'b01: Dedicated full-speed interface</li> <li>▪ 2'b10: FS pins shared with UTMI+ pins</li> <li>▪ 2'b11: FS pins shared with ULPI pins</li> </ul> <p>This field is configured using parameter "Name: <a href="#">OTG_FSPHY_INTERFACE</a>".</p> | Configurable | RO |
| 7:6   | <p>High-Speed PHY Interface Type (HSPhyType)</p> <ul style="list-style-type: none"> <li>▪ 2'b00: High-Speed interface not supported</li> <li>▪ 2'b01: UTMI+</li> <li>▪ 2'b10: ULPI</li> <li>▪ 2'b11: UTMI+and ULPI</li> </ul> <p>This field is configured using parameter "Name: <a href="#">OTG_HSPHY_INTERFACE</a>".</p>                                                                    | Configurable | RO |
| 5     | <p>Point-to-Point (SingPnt)</p> <p>1'b0: Multi-point application (hub and split support)</p> <p>1'b1: Single-point application (no hub and no split support)</p> <p>This field is configured using parameter "Name: <a href="#">OTG_SINGLE_POINT</a>".</p>                                                                                                                                    | Configurable | RO |
| 4:3   | <p>Architecture (OtgArch)</p> <ul style="list-style-type: none"> <li>▪ 2'b00: Slave-Only</li> <li>▪ 2'b01: External DMA</li> <li>▪ 2'b10: Internal DMA</li> <li>▪ Others: Reserved</li> </ul> <p>This field is configured using parameter "Name: <a href="#">OTG_ARCHITECTURE</a>".</p>                                                                                                       | Configurable | RO |
| 2:0   | <p>Mode of Operation (OtgMode)</p> <ul style="list-style-type: none"> <li>▪ 3'b000: HNP- and SRP-Capable OTG (Host and Device)</li> <li>▪ 3'b001: SRP-Capable OTG (Host and Device)</li> <li>▪ 3'b010: Non-HNP and Non-SRP Capable OTG (Host and Device)</li> <li>▪ 3'b011: SRP-Capable Device</li> <li>▪ 3'b100: Non-OTG Device</li> </ul>                                                   | Configurable | RO |

|  |                                                                                                                                                                                                            |  |  |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | <ul style="list-style-type: none"> <li>▪ 3'b101: SRP-Capable Host</li> <li>▪ 3'b110: Non-OTG Host</li> <li>▪ Others: Reserved</li> </ul> <p>This field is configured using parameter "Name: OTG_MODE".</p> |  |  |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

### 33.5.3.14 User HW Config3 Register (GHWCFG3)

◇ Offset: 04Ch

This register contains the configuration options selected.

**Table 33-22 User HW Config3 Register: GHWCFG3**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                   | Reset        | Access |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31:16 | DFIFO Depth (DfifoDepth minus EP_LOC_CNT)<br>This value is in terms of 32-bit words. <ul style="list-style-type: none"> <li>▪ Minimum value is 32</li> <li>▪ Maximum value is 32,768</li> </ul> This field is configured using parameter "Name: OTG_DFIFO_DEPTH" and EP_LOC_CNT. For more information on EP_LOC_CNT, see "Endpoint Information Controller (EPINFO_CTL)".                                      | Configurable | RO     |
| 15    | OTG_ENABLE_LPM<br>LPM mode specified for Mode of Operation (parameter OTG_ENABLE_LPM) in coreConsultant configuration.                                                                                                                                                                                                                                                                                        | Configurable | RO     |
| 14    | OTG_BC_SUPPORT<br>This bit indicates the HS OTG controller support for Battery Charger. <ul style="list-style-type: none"> <li>▪ 0 - No Battery Charger Support</li> <li>▪ 1 - Battery Charger support present.</li> </ul>                                                                                                                                                                                    | Configurable | RO     |
| 13    | OTG_ENABLE_HSIC<br>HSIC mode specified for Mode of Operation in coreConsultant configuration (parameter OTG_ENABLE_HSIC).<br><b>Value Range:</b> 0-1 <ul style="list-style-type: none"> <li>▪ 1: HSIC-capable with shared UTMI PHY interface</li> <li>▪ 0: Non-HSIC-capable</li> </ul> <b>Parameter Name:</b> OTG_ENABLE_HSIC<br><b>Dependencies:</b> OTG_HSPHY_INTERFACE<br><b>Default:</b> Non-HSIC-capable |              | RO     |
| 12    | OTG_ADP_SUPPORT<br>This bit indicates whether ADP logic is present within or external to the HS OTG controller <ul style="list-style-type: none"> <li>▪ 0 - No ADP logic present with HS OTG controller</li> <li>▪ 1- ADP logic is present along with HS OTG controller.</li> </ul>                                                                                                                           |              | RO     |
| 11    | Reset Style for Clocked always Blocks in RTL (RstType)                                                                                                                                                                                                                                                                                                                                                        | Configurable | RO     |



|     |                                                                                                                                                                                                                                                                                                                                                                                                                                        |              |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|----|
|     | <ul style="list-style-type: none"> <li>▪ 1'b0: Asynchronous reset is used in the core</li> <li>▪ 1'b1: Synchronous reset is used in the core</li> </ul> This field is configured using parameter " <a href="#">Name: OTG_SYNC_RESET_TYPE</a> ".                                                                                                                                                                                        |              |    |
| 10  | Optional Features Removed (OptFeature)<br>Indicates whether the User ID register, GPIO interface ports, and SOF toggle and counter ports were removed for gate count optimization by enabling Remove Optional Features? during coreConsultant configuration. <ul style="list-style-type: none"> <li>▪ 1'b0: No</li> <li>▪ 1'b1: Yes</li> </ul> This field is configured using parameter " <a href="#">Name: OTG_RM_OPT_FEATURES</a> ". | Configurable | RO |
| 9   | Vendor Control Interface Support (VndctlSupt) <ul style="list-style-type: none"> <li>▪ 1'b0: Vendor Control Interface is not available on the core.</li> <li>▪ 1'b1: Vendor Control Interface is available.</li> </ul> This field is configured using parameter " <a href="#">Name: OTG_VENDOR_CTL_INTERFACE</a> ".                                                                                                                    | Configurable | RO |
| 8   | I <sup>2</sup> C Selection (I2CIntSel) <ul style="list-style-type: none"> <li>▪ 1'b0: I 2C Interface is not available on the core.</li> <li>▪ 1'b1: I 2C Interface is available on the core.</li> </ul> This field is configured using parameter " <a href="#">Name: OTG_I2C_INTERFACE</a> ".                                                                                                                                          | Configurable | RO |
| 7   | OTG Function Enabled (OtgEn)<br>The application uses this bit to indicate the DWC_otg core's OTG capabilities. <ul style="list-style-type: none"> <li>▪ 1'b0: Not OTG capable</li> <li>▪ 1'b1: OTG Capable</li> </ul> This field is configured using parameter " <a href="#">Name: OTG_MODE</a> ".                                                                                                                                     | Configurable | RO |
| 6:4 | Width of Packet Size Counters (PktSizeWidth) <ul style="list-style-type: none"> <li>▪ 3'b000: 4 bits</li> <li>▪ 3'b001: 5 bits</li> <li>▪ 3'b010: 6 bits</li> <li>▪ 3'b011: 7 bits</li> <li>▪ 3'b100: 8 bits</li> <li>▪ 3'b101: 9 bits</li> <li>▪ 3'b110: 10 bits</li> <li>▪ Others: Reserved</li> </ul> This field is configured using parameter " <a href="#">Name: OTG_PACKET_COUNT_WIDTH</a> ".                                    | Configurable | RO |
| 3:0 | Width of Transfer Size Counters (XferSizeWidth) <ul style="list-style-type: none"> <li>▪ 4'b0000: 11 bits</li> <li>▪ 4'b0001: 12 bits</li> </ul>                                                                                                                                                                                                                                                                                       | Configurable | RO |

|  |                                                                                                                                                                                               |  |  |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | ... <ul style="list-style-type: none"> <li>▪ 4'b1000: 19 bits</li> <li>▪ Others: Reserved</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_TRANS_COUNT_WIDTH</a> ". |  |  |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

### 33.5.3.15 User HW Config4 Register (GHWCFG4)

◇ Offset: 050h

This register contains the configuration options selected.

Note: Bit [31] is available only when Scatter/Gather DMA mode is enabled. When Scatter/Gather DMA mode is disabled, this field is reserved.

**Table 33-23 User HW Config4 Register: GHWCFG4**

| Field | Description                                                                                                                                                                                                                                                                                                                     | Reset        | Access |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31    | Scatter/Gather DMA <ul style="list-style-type: none"> <li>▪ 1'b1: Dynamic configuration</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_EN_DESC_DMA</a> ".                                                                                                                                           | Configurable | RO     |
| 30    | Scatter/Gather DMA configuration <ul style="list-style-type: none"> <li>▪ 1'b0: Non-Scatter/Gather DMA configuration</li> <li>▪ 1'b1: Scatter/Gather DMA configuration</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_EN_DESC_DMA</a> ".                                                            | Configurable | RO     |
| 29:26 | Number of Device Mode IN Endpoints Including Control Endpoints (INEps) Range 0 -15 <ul style="list-style-type: none"> <li>▪ 0:1 IN Endpoint</li> <li>▪ 1:2 IN Endpoints</li> <li>....</li> <li>▪ 15:16 IN Endpoints</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_NUM_IN_EPS</a> ".                | Configurable | RO     |
| 25    | Enable Dedicated Transmit FIFO for device IN Endpoints (DedFifoMode) <ul style="list-style-type: none"> <li>▪ 1'b0: Dedicated Transmit FIFO Operation not enabled.</li> <li>▪ 1'b1: Dedicated Transmit FIFO Operation enabled.</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_EN_DED_TX_FIFO</a> ". | Configurable | RO     |
| 24    | session_end Filter Enabled (SessEndFltr) <ul style="list-style-type: none"> <li>▪ 1'b0: No filter</li> <li>▪ 1'b1: Filter</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_EN_SESSIONEND_FILTER</a> ".                                                                                                | Configurable | RO     |

|       |                                                                                                                                                                                                                                                                                                                                                                                                      |              |    |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|----|
| 23    | "b_valid" Filter Enabled (BValidFtr) <ul style="list-style-type: none"> <li>▪ 1'b0: No filter</li> <li>▪ 1'b1: Filter</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_EN_B_VALID_FILTER</a> ".                                                                                                                                                                            | Configurable | RO |
| 22    | "a_valid" Filter Enabled (AValidFtr) <ul style="list-style-type: none"> <li>▪ 1'b0: No filter</li> <li>▪ 1'b1: Filter</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_EN_A_VALID_FILTER</a> ".                                                                                                                                                                            | Configurable | RO |
| 21    | "vbus_valid" Filter Enabled (VBusValidFtr) <ul style="list-style-type: none"> <li>▪ 1'b0: No filter</li> <li>▪ 1'b1: Filter</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_EN_VBUSVALID_FILTER</a> ".                                                                                                                                                                    | Configurable | RO |
| 20    | "iddig" Filter Enable (IddgFtr) <ul style="list-style-type: none"> <li>▪ 1'b0: No filter</li> <li>▪ 1'b1: Filter</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_EN_IDDIG_FILTER</a> ".                                                                                                                                                                                   | Configurable | RO |
| 19:16 | Number of Device Mode Control Endpoints in Addition to Endpoint 0 (NumCtlEps)<br>Range: 0-15<br>This field is configured using parameter "Name: <a href="#">OTG_NUM_CRL_EPS</a> ".                                                                                                                                                                                                                   | Configurable | RO |
| 15:14 | UTMI+ PHY/ULPI-to-Internal UTMI+ Wrapper Data Width (PhyDataWidth) When a ULPI PHY is used, an internal wrapper converts ULPI to UTMI+. <ul style="list-style-type: none"> <li>▪ 2'b00: 8 bits</li> <li>▪ 2'b01: 16 bits</li> <li>▪ 2'b10: 8/16 bits, software selectable</li> <li>▪ Others: Reserved</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_HSPHY_DWIDTH</a> ". | Configurable | RO |
| 13:6  | Reserved                                                                                                                                                                                                                                                                                                                                                                                             |              | RO |
| 6     | Enable Hibernation <ul style="list-style-type: none"> <li>▪ 1'b0: Hibernation feature not enabled</li> <li>▪ 1'b1: Hibernation feature enabled</li> </ul> This field is configured using parameter "Name: <a href="#">OTG_EN_PWROPT</a> ".                                                                                                                                                           | Configurable | RO |
| 5     | Minimum AHB Frequency Less Than 60 MHz (AhbFreq) <ul style="list-style-type: none"> <li>▪ 1'b0: No</li> <li>▪ 1'b1: Yes</li> </ul> This field is configured using parameter "Name:                                                                                                                                                                                                                   | Configurable | RO |

|     |                                                                                                                                                                                                                                                  |              |    |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|----|
|     | <a href="#">OTG_MIN_AHB_FREQ_LESSTHAN_60"</a> .                                                                                                                                                                                                  |              |    |
| 4   | Enable Partial Power Down <ul style="list-style-type: none"> <li>▪ 1'b0: Partial Power Down Not Enabled</li> <li>▪ 1'b1: Partial Power Down Enabled</li> </ul> This field is configured using parameter " <a href="#">Name: OTG_EN_PWROPT"</a> . | Configurable | RO |
| 3:0 | Number of Device Mode Periodic IN Endpoints (NumDevPerioEps)<br>Range: 0-15<br>This field is configured using parameter " <a href="#">Name: OTG_NUM_PERIO_EPS"</a> .                                                                             | Configurable | RO |

### 33.5.3.16 DFIFO Software Config Register (GDFIFOCFG)

✧ Offset: 05Ch

**Table 33-24 Global DFIFO Software Config Register: GDFIFOCFG**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                               | Mode            | Reset               | Acces |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------------------|-------|
| 31:16 | EPInfoBaseAddr<br>This field provides the start address of the EP info controller.                                                                                                                                                                                                                                                                                                        | Host and Device | EPINFO_BA<br>SEADDR | R_W   |
| 15:0  | GDFIFOCfg<br>This field is for dynamic programming of the DFIFO Size. This value takes effect only when the application programs a non zero value to this register. The value programmed must conform to the guidelines described in "FIFO RAM Allocation" in the Programming Guide.<br>The DWC_otg core does not have any corrective logic if the FIFO sizes are programmed incorrectly. | Host and Device | OTG_DFIFO<br>_DEPTH | R_W   |

### 33.5.3.17 Host Periodic Transmit FIFO Size Register (HPTXFSIZ)

✧ Offset: 100h

This register holds the size and the memory start address of the Periodic TxFIFO.

**Table 33-25 Host Periodic Transmit FIFO Size Register: HPTXFSIZ**

| Field | Description                                                                                                                                                                               | Reset        | Acces  |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31:16 | Host Periodic TxFIFO Depth (PTxFSize)<br>This value is in terms of 32-bit words. <ul style="list-style-type: none"> <li>▪ Minimum value is 16</li> <li>▪ Maximum value is 3648</li> </ul> | Configurable | R_W    |
| 15:0  | Host Periodic TxFIFO Start Address (PTxFStAddr)<br>The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth and Largest Non-periodic Tx Data FIFO Depth     | Configurable | RO/R_W |

|  |                                                                                                                                                                                                                                                                                                                                |  |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | specified .These parameters are:<br>In shared FIFO operation:- <ul style="list-style-type: none"> <li>▪ OTG_RX_DFIFO_DEPTH(3648) +<br/>OTG_TX_NPERIO_DFIFO_DEPTH(1024)</li> </ul> In dedicated FIFO mode:- <ul style="list-style-type: none"> <li>▪ OTG_RX_DFIFO_DEPTH(3648) +<br/>OTG_TX_HNPERIO_DFIFO_DEPTH(3648)</li> </ul> |  |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

### 33.5.3.18 Device Periodic Transmit FIFO-*n* Size Register (DPTXFSIZn)

◇ FIFO\_number:  $1 \leq n \leq 15$

◇ Offset:  $104h + (FIFO\_number - 1) * 04h$

This register is valid only in shared FIFO operation (OTG\_EN\_DED\_TX\_FIFO = 0).

This register holds the memory start address of each periodic Tx FIFO to be implemented in Device mode. Each periodic FIFO holds the data for one periodic IN endpoint. This register is repeated for each periodic FIFO instantiated.

**Table 33-26 Device Periodic Transmit FIFO-*n* Register: DPTXFSIZn**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset        | Access |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31:16 | Device Periodic Tx FIFO Size (DPTxFSize)<br>This value is in terms of 32-bit words. <ul style="list-style-type: none"> <li>▪ Minimum value is 4</li> <li>▪ Maximum value is 768</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Configurable | RO     |
| 15:0  | Device Periodic Tx FIFO RAM Start Address (DPTxFStAddr)<br>This field specifies the start address in the RAM for this periodic FIFO.<br>The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth, Largest Non-periodic Tx Data FIFO Depth, and all lower numbered Largest Device Mode Periodic Tx Data FIFO Depth specified .<br>The formula used is:<br>$OTG\_RX\_DFIFO\_DEPTH + OTG\_TX\_NPERIO\_DFIFO\_DEPTH +$<br>$SUM\ of\ OTG\_TX\_DPERIO\_DFIFO\_DEPTH\_x' (where\ x=1\ to\ n-1).$<br>When $n = 1$ , the above expression becomes<br>$OTG\_RX\_DFIFO\_DEPTH + OTG\_TX\_NPERIO\_DFIFO\_DEPTH.$<br>If at POR, the calculated value (C) exceeds 65535, then the Reset value becomes Reset Value(A) = (C - 65536). | Configurable | RO/R_W |

### 33.5.3.19 Device IN Endpoint Transmit FIFO Size Register: (DIEPTxFn)

◇ FIFO\_number:  $1 \leq n \leq 15$

◇ Offset: 104h + (FIFO\_number - 1) \* 04h

This register is valid only in dedicated FIFO mode.

This register holds the size and memory start address of IN endpoint TxFIFOs implemented in Device mode. Each FIFO holds the data for one IN endpoint. This register is repeated for instantiated IN endpoint FIFOs 1 to 15. For IN endpoint FIFO 0 use GNPTXFSIZ register for programming the size and memory start address.

**Table 33-27 Device In Endpoint Transmit FIFO Size Register: (DIEPTXFn)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset        | Access |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31:16 | IN Endpoint TxFIFO Depth (INEPnTxFDep)<br>This value is in terms of 32-bit words.<br>Minimum value is 16<br>Maximum value is 3648                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Configurable | R_W    |
| 15:0  | IN Endpoint FIFO <sub>n</sub> Transmit RAM Start Address (INEPnTxFStAddr)<br>This field contains the memory start address for IN endpoint Transmit FIFO <sub>n</sub> (0 < n ≤ 15). The power-on reset value of this register is calculated according to the following formula:<br>OTG_RX_DFIFO_DEPTH + SUM of<br>OTG_TX_DINEP_DFIFO_DEPTH_'x' (where x = 0 to n - 1)<br>If at POR the calculated value (C) exceeds 65535, then the Reset value becomes Reset Value(A) = (C - 65536).<br>Example:<br>If start address of IN endpoint FIFO 1 is OTG_RX_DFIFO_DEPTH + OTG_TX_DINEP_DFIFO_DEPTH_0 and start address of IN endpoint FIFO 2 is OTG_RX_DFIFO_DEPTH + OTG_TX_DINEP_DFIFO_DEPTH_0 + OTG_TX_DINEP_DFIFO_DEPTH_1. | Configurable | RO/R_W |

### 33.5.4 Host Mode Registers

These registers affect the operation of the core in the Host mode. Host mode registers must not be accessed in Device mode, as the results are undefined. Host Mode registers can be categorized as follows:

#### 33.5.4.1 Host Configuration Register (HCFG)

◇ Offset: 400h

This register configures the core after power-on. Do not make changes to this register after initializing the host.

**Table 33-28 Host Configuration Register: HCFG**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset | Access |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| 31    | Mode Change Ready Timer Enable (ModeChTimEn)<br>This bit is used to enable or disable the host core to wait for 200 PHY clock cycles at the end of Resume to change the opmode signal to the PHY to 00 after Suspend or LPM. <ul style="list-style-type: none"> <li>▪ 1'b0: The Host core waits for either 200 PHY clock cycles or a linestate of SE0 at the end of resume to change the opmode from 2'b10 to 2'b00.</li> <li>▪ 1'b1: The Host core waits only for a linestate of SE0 at the end of resume to change the opmode from 2'b10 to 2'b00.</li> </ul>                                                                                                                                                           | 1'b0  | R_W    |
| 30:27 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |       | RO     |
| 26    | Enable Periodic Scheduling (PerSchedEna)<br>Applicable in Scatter/Gather DMA mode only. Enables periodic scheduling within the core. Initially, the bit is reset. The core will not process any periodic channels. As soon as this bit is set, the core will get ready to start scheduling periodic channels. In non Scatter/Gather DMA mode, this bit is reserved.                                                                                                                                                                                                                                                                                                                                                       | 1'b0  | R_W    |
| 25:24 | Frame List Entries (FrListEn). The value in the register specifies the number of entries in the Frame list. This field is valid only in Scatter/Gather DMA mode. <ul style="list-style-type: none"> <li>▪ 2'b00: 8 Entries</li> <li>▪ 2'b01: 16 Entries</li> <li>▪ 2'b10: 32 Entries</li> <li>▪ 2'b11: 64 Entries</li> </ul> In modes other than Scatter/Gather DMA mode, these bits are reserved.                                                                                                                                                                                                                                                                                                                        | 2'b00 | R/W    |
| 23    | Enable Scatter/gather DMA in Host mode (DescDMA)<br>When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation.<br>NOTE: This bit must be modified only once after a reset. The following combinations are available for programming: <ul style="list-style-type: none"> <li>▪ GAHBCFG.DMAEn=0, HCFG.DescDMA=0 =&gt; Slave mode</li> <li>▪ GAHBCFG.DMAEn=0, HCFG.DescDMA=1 =&gt; Invalid</li> <li>▪ GAHBCFG.DMAEn=1, HCFG.DescDMA=0 =&gt; Buffered DMA mode</li> <li>▪ GAHBCFG.DMAEn=1, HCFG.DescDMA=1 =&gt; Scatter/Gather DMA mode</li> </ul> In non Scatter/Gather DMA mode, this bit is reserved. | 1'b0  | R/W    |
| 22:16 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 8'd2  | R_W    |
| 15:8  | Resume Validation Period (ResValid)<br>This field is effective only when HCFG.Ena32KHzS is set. It controls the resume period when the core resumes from suspend. The core                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 8'd2  | R_W    |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |     |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
|     | counts the ResValid number of clock cycles to detect a valid resume when this is set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |      |     |
| 7   | <p>Enable 32-KHz Suspend Mode (Ena32KHzS)</p> <p>This bit can only be set if the USB 1.1 Full-Speed Serial Transceiver Interface has been selected. If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero. When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1'd0 | R_W |
| 6:3 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      | RO  |
| 2   | <p>FS- and LS-Only Support (FSLSSupp)</p> <p>The application uses this bit to control the core's enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device</li> <li>▪ 1'b1: FS/LS-only, even if the connected device can support HS</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 1'b0 | R_W |
| 1:0 | <p>FS/LS PHY Clock Select (FSLSPclkSel)</p> <p>When the core is in FS Host mode</p> <ul style="list-style-type: none"> <li>▪ 2'b00: Internal PHY clock is running at 30/60 MHz (For UTMI+/ULPI PHY Interfaces)</li> <li>▪ 2'b01: Internal PHY clock is running at 48MHz (For 1.1 FS transceiver Interface)</li> <li>▪ Others: Reserved</li> </ul> <p>When the core is in LS Host mode</p> <ul style="list-style-type: none"> <li>▪ 2'b00: Internal PHY clock is running at 30/60 MHz (For UTMI+/ULPI PHY Interfaces)</li> <li>▪ 2'b10: Internal PHY clock is running at 6 MHz and the external clock is running at 48MHz. When you select a 6 MHz clock during LS Mode, you must do a soft reset (for 1.1 FS transceiver Interface)</li> <li>▪ Others: Reserved</li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>▪ When Core in FS mode, the internal and external clocks have the same frequency.</li> <li>▪ When Core in LS mode, <ul style="list-style-type: none"> <li>– If FSLSPclkSel = 2'b00: Internal and external clocks have the same frequency</li> <li>– If FSLSPclkSel = 2'b10: Internal clock is divided by eight version of external 48 MHz clock (utmifs_clk).</li> </ul> </li> </ul> | 2'b0 | R_W |



### 33.5.4.2 Host Frame Interval Register (HFIR)

◇ Offset: 404h

This register stores the frame interval information for the current speed to which the DWC\_otg core has enumerated.

**Table 33-29 Host Frame Interval Register: HFIR**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset     | Access |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------|
| 31:17 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |           | RO     |
| 16    | Reload Control (HFIRRldCtrl)<br>This bit allows dynamic reloading of the HFIR register during runtime. <ul style="list-style-type: none"> <li>▪ 1'b0: The HFIR cannot be reloaded dynamically</li> <li>▪ 1'b1: the HFIR can be dynamically reloaded during runtime.</li> </ul> This bit needs to be programmed during initial configuration and its value must not be changed during runtime.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 1'b0      | R_W    |
| 15:0  | Frame Interval (FrInt)<br>The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro-SOFs (HS) or Keep-Alive tokens (LS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for a FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). <ul style="list-style-type: none"> <li>▪ 125 <math>\mu</math>s * (PHY clock frequency for HS)</li> <li>▪ 1 ms * (PHY clock frequency for FS/LS)</li> </ul> | 16'd60000 | R_W    |

### 33.5.4.3 Host Frame Number/Frame Time Remaining Register (HFNUM)

◇ Offset: 408h

This register indicates the current frame number. It also indicates the time remaining (in terms of the number of PHY clocks) in the current (micro)frame.

**Table 33-30 Host Frame Number/Frame Time Remaining Register: HFNUM**

| Field | Description                                                                                                                                                                                                                                                                                                         | Reset | Access |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| 31:16 | Frame Time Remaining (FrRem)<br>Indicates the amount of time remaining in the current microframe (HS) or frame (FS/LS), in terms of PHY clocks. This field decrements on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on | 16'h0 | RO     |

|      |                                                                                                                                            |          |    |
|------|--------------------------------------------------------------------------------------------------------------------------------------------|----------|----|
|      | the USB.                                                                                                                                   |          |    |
| 15:0 | <b>Frame Number (FrNum)</b><br>This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF. | 16'h3FFF | RO |

### 33.5.4.4 Host Periodic Transmit FIFO/Queue Status Register (HPTXSTS)

◇ Offset: 410h

This read-only register contains the free space information for the Periodic TxFIFO and the Periodic Transmit Request Queue.

**Table 33-31 Host Periodic Transmit FIFO/Queue Status Register: HPTXSTS**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset        | Access |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31:24 | <b>Top of the Periodic Transmit Request Queue (PTxQTop)</b><br>This indicates the entry in the Periodic Tx Request Queue that is currently being processed by the MAC.<br>This register is used for debugging. <ul style="list-style-type: none"> <li>▪ Bit [31]: Odd/Even (micro)frame               <ul style="list-style-type: none"> <li>– 1'b0: send in even (micro)frame</li> <li>– 1'b1: send in odd (micro)frame</li> </ul> </li> <li>▪ Bits [30:27]: Channel/endpoint number</li> <li>▪ Bits [26:25]: Type               <ul style="list-style-type: none"> <li>– 2'b00: IN/OUT</li> <li>– 2'b01: Zero-length packet</li> <li>– 2'b10: CSPLIT</li> <li>– 2'b11: Disable channel command</li> </ul> </li> <li>▪ Bit [24]: Terminate (last entry for the selected channel/endpoint)</li> </ul> | 8'h0         | RO     |
| 23:16 | <b>Periodic Transmit Request Queue Space Available (PTxQSpcAvail)</b><br>Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. <ul style="list-style-type: none"> <li>▪ 8'h0: Periodic Transmit Request Queue is full</li> <li>▪ 8'h1: 1 location available</li> <li>▪ 8'h2: 2 locations available</li> <li>▪ n : n locations available (0 ≤ n ≤ 16)</li> <li>▪ Others: Reserved</li> </ul>                                                                                                                                                                                                                                                                                                               | Configurable | RO     |
| 15:0  | <b>Periodic Transmit Data FIFO Space Available (PTxFSpcAvail)</b><br>Indicates the number of free locations available to be written to in the Periodic TxFIFO.<br>Values are in terms of 32-bit words <ul style="list-style-type: none"> <li>▪ 16'h0: Periodic TxFIFO is full</li> <li>▪ 16'h1: 1 word available</li> <li>▪ 16'h2: 2 words available</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                       | Configurable | RO     |

|  |                                                                                                                                                                                                |  |  |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | <ul style="list-style-type: none"> <li>▪ 16'hn: n words available (where <math>0 \leq n \leq 32,768</math>)</li> <li>▪ 16'h8000: 32,768 words available</li> <li>▪ Others: Reserved</li> </ul> |  |  |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

### 33.5.4.5 Host All Channels Interrupt Register (HAINT)

◇ Offset: 414h

When a significant event occurs on a channel, the Host All Channels Interrupt register interrupts the application using the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt). This is shown in Figure 33-2. There is one interrupt bit per channel, up to a maximum of 16 bits. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Host Channel-n Interrupt register.

**Table 33-32 Host All Channels Interrupt Register: HAINT**

| Field | Description                                                                                   | Reset | Access |
|-------|-----------------------------------------------------------------------------------------------|-------|--------|
| 31:16 | Reserved                                                                                      |       | RO     |
| 15:0  | Channel Interrupts (HAINT)<br>One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15 | 16'h0 | RO     |

### 33.5.4.6 Host All Channels Interrupt Mask Register (HAINTMSK)

◇ Offset: 418h

The Host All Channel Interrupt Mask register works with the Host All Channel Interrupt register to interrupt the application when an event occurs on a channel. There is one interrupt mask bit per channel, up to a maximum of 16 bits.

◇ Mask interrupt: 1'b0

◇ Unmask interrupt: 1'b1

**Table 33-33 Host All Channels Interrupt Mask Register: HAINTMSK**

| Field | Description                                                                                          | Reset | Access |
|-------|------------------------------------------------------------------------------------------------------|-------|--------|
| 31:16 | Reserved                                                                                             |       | RO     |
| 15:0  | Channel Interrupt Mask (HAINTMsk)<br>One bit per channel: Bit 0 for channel 0, bit 15 for channel 15 | 16'h0 | R_W    |

### 33.5.4.7 Host Port Control and Status Register (HPRT)

◇ Offset: 440h

This register is available only in Host mode. Currently, the OTG Host supports only one port.

A single register holds USB port-related information such as USB reset, enable, suspend, resume, connect status, and test mode for each port. It is shown in Figure 33-2. The R\_SS\_WC bits in this register can trigger an interrupt to the application through the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). On a Port Interrupt, the application must read this register and

clear the bit that caused the interrupt. For the R\_SS\_WC bits, the application must write a 1 to the bit to clear the interrupt.

**Table 33-34 Host Port Control and Status Register: HPRT**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset | Access |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| 31:19 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       | RO     |
| 18:17 | Port Speed (PrtSpd)<br>Indicates the speed of the device attached to this port. <ul style="list-style-type: none"> <li>▪ 2'b00: High speed</li> <li>▪ 2'b01: Full speed</li> <li>▪ 2'b10: Low speed</li> <li>▪ 2'b11: Reserved</li> </ul>                                                                                                                                                                                                                                                                                                                                                                            | 2'h0  | RO     |
| 16:13 | Port Test Control (PrtTstCtl)<br>The application writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is signaled on the port. <ul style="list-style-type: none"> <li>▪ 4'b0000: Test mode disabled</li> <li>▪ 4'b0001: Test_J mode</li> <li>▪ 4'b0010: Test_K mode</li> <li>▪ 4'b0011: Test_SE0_NAK mode</li> <li>▪ 4'b0100: Test_Packet mode</li> <li>▪ 4'b0101: Test_Force_Enable</li> <li>▪ Others: Reserved</li> </ul> For more information on moving the DWC_otg host core into test mode, see <a href="#">"Moving the Host Core to Test Mode"</a> on page 89. | 4'h0  | R_W    |
| 12    | Port Power (PrtPwr)<br>The application uses this field to control power to this port (write 1'b1 to set to 1'b1 and write 1'b0 to set to 1'b0), and the core can clear this bit on an over current condition. <ul style="list-style-type: none"> <li>▪ 1'b0: Power off</li> <li>▪ 1'b1: Power on</li> </ul>                                                                                                                                                                                                                                                                                                          | 1'b0  | RO     |
| 11:10 | Port Line Status (PrtLnSts)<br>Indicates the current logic level USB data lines <ul style="list-style-type: none"> <li>▪ Bit [10]: Logic level of D+</li> <li>▪ Bit [11]: Logic level of D-</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                               | 2'b0  | RO     |
| 9     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       | RO     |
| 8     | Port Reset (PrtRst)<br>When the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. <ul style="list-style-type: none"> <li>▪ 1'b0: Port not in reset</li> <li>▪ 1'b1: Port in reset</li> </ul> To start a reset on the port, the application must leave this bit set                                                                                                                                                                                                                            | 1'b0  | R_W    |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |           |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----------|
|   | <p>for at least the minimum duration mentioned below, as specified in the USB 2.0 specification, Section 7.1.7.5. The application can leave it set for another 10 ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit set by the USB standard.</p> <ul style="list-style-type: none"> <li>▪ High speed: 50 ms</li> <li>▪ Full speed/Low speed: 10 ms</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |      |           |
| 7 | <p>Port Suspend (PrtSusp)</p> <p>The application sets this bit to put this port in Suspend mode. The core only stops sending SOFs when this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the suspend input pin of the PHY.</p> <p>The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.WkUpInt or GINTSTS.DisconnInt, respectively).</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Port not in Suspend mode</li> <li>▪ 1'b1: Port in Suspend mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                   | 1'b0 | R_WS_SC   |
| 6 | <p>Port Resume (PrtRes)</p> <p>The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit.</p> <p>If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/Remote Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.WkUpInt), the core starts driving resume signaling without application intervention and clears this bit when it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: No resume driven</li> <li>▪ 1'b1: Resume driven</li> </ul> <p>When LPM is enabled and the core is in the L1 (Sleep) state, setting this bit results in the following behavior:</p> <p>The core continues to drive the resume signal until a pre-determined time specified in the GLPMCFG.HIRD_Thres[3:0] field.</p> <p>If the core detects a USB remote wakeup sequence, as indicated by the Port L1 Resume/Remote L1 Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.L1WkUpInt), the core starts driving resume signaling without application intervention</p> | 1'b0 | R_W_SS_SC |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |      |            |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------------|
|   | and clears this bit at the end of the resume. The read value of this bit indicates whether the core is currently driving resume signaling. <ul style="list-style-type: none"> <li>▪ 1'b0: No resume driven</li> <li>▪ 1'b1: Resume driven</li> </ul>                                                                                                                                                                                                                     |      |            |
| 5 | Port Overcurrent Change (PrtOvrCurrChng)<br>The core sets this bit when the status of the Port Overcurrent Active bit (bit 4) in this register changes.                                                                                                                                                                                                                                                                                                                  | 1'b0 | R_SS_WC    |
| 4 | Port Overcurrent Active (PrtOvrCurrAct)<br>Indicates the overcurrent condition of the port. <ul style="list-style-type: none"> <li>▪ 1'b0: No overcurrent condition</li> <li>▪ 1'b1: Overcurrent condition</li> </ul>                                                                                                                                                                                                                                                    | 1'b0 | RO         |
| 3 | Port Enable/Disable Change (PrtEnChng)<br>The core sets this bit when the status of the Port Enable bit [2] of this register changes.                                                                                                                                                                                                                                                                                                                                    | 1'b0 | R_SS_WC    |
| 2 | Port Enable (PrtEna)<br>A port is enabled only by the core after a reset sequence, and is disabled by an overcurrent condition, a disconnect condition, or by the application clearing this bit.<br>The application cannot set this bit by a register write. It can only clear it to disable the port. This bit does not trigger any interrupt to the application. <ul style="list-style-type: none"> <li>▪ 1'b0: Port disabled</li> <li>▪ 1'b1: Port enabled</li> </ul> | 1'b0 | R_SS_SC_WC |
| 1 | Port Connect Detected (PrtConnDet)<br>The core sets this bit when a device connection is detected to trigger an interrupt to the application using the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). The application must write a 1 to this bit to clear the interrupt.                                                                                                                                                                       | 1'b0 | R_SS_WC    |
| 0 | Port Connect Status (PrtConnSts) <ul style="list-style-type: none"> <li>▪ 0: No device is attached to the port.</li> <li>▪ 1: A device is attached to the port.</li> </ul>                                                                                                                                                                                                                                                                                               | 1'b0 | RO         |

#### 33.5.4.7.1 Moving the Host Core to Test Mode

To move the DWC\_otg core to test mode, you must set HPRT.Port Test Control. Complete the following steps to move the DWC\_otg core to test mode:

1. Power on the core.
2. Load the DWC\_otg driver.
3. Connect an HS device and enumerate to HS mode.
4. Access the HPRT register to send test packets.
5. Remove the device and connect to fixture (OPT) port.

The DWC\_otg host core continues sending out test packets.

6. Test the eye diagram.

### 33.5.4.8 Host Channel-*n* Characteristics Register (HCCHAR<sub>n</sub>)

- ✧ Channel\_number:  $0 \leq n \leq 15$
- ✧ Offset:  $500h + (\text{Channel\_number} * 20h)$

**Table 33-35 Host Channel-*n* Characteristics Register: HCCHAR<sub>n</sub>**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset | Access     |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|
| 31    | Channel Enable (ChEna)<br>When Scatter/Gather mode is enabled <ul style="list-style-type: none"> <li>▪ 1'b0: Indicates that the descriptor structure is not yet ready.</li> <li>▪ 1'b1: Indicates that the descriptor structure and data buffer with data is setup and this channel can access the descriptor.</li> </ul> When Scatter/Gather mode is disabled<br>This field is set by the application and cleared by the OTG host. <ul style="list-style-type: none"> <li>▪ 1'b0: Channel disabled</li> <li>▪ 1'b1: Channel enabled</li> </ul> | 1'b0  | R_WS_SC    |
| 30    | Channel Disable (ChDis)<br>The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.                                                                                                                                                                                                                                                                         | 1'b0  | R_WS_SC_SS |
| 29    | Odd Frame (OddFrm)<br>This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro)frame. This field is applicable for only periodic (isochronous and interrupt) transactions. <ul style="list-style-type: none"> <li>▪ 1'b0: Even (micro)frame</li> <li>▪ 1'b1: Odd (micro)frame</li> </ul> This field is not applicable for Scatter/Gather DMA mode and need not be programmed by the application and is ignored by the core.                                                          | 1'b0  | R_W        |
| 28:22 | Device Address (DevAddr)<br>This field selects the specific device serving as the data source or sink.                                                                                                                                                                                                                                                                                                                                                                                                                                          | 7'h0  | R_W        |
| 21:20 | Multi Count (MC) / Error Count (EC)<br>When the Split Enable bit of the Host Channel- <i>n</i> Split Control register (HCSPLT <sub>n</sub> .SplitEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this periodic endpoint. For non periodic transfers,                                                                                                                                                                                                                 | 2'b0  | R_W        |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       |     |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|
|       | <p>this field is used only in DMA mode, and specifies the number packets to be fetched for this channel before the internal DMA engine changes arbitration.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: Reserved This field yields undefined results.</li> <li>▪ 2'b01: 1 transaction</li> <li>▪ 2'b10: 2 transactions to be issued for this endpoint per microframe</li> <li>▪ 2'b11: 3 transactions to be issued for this endpoint per microframe</li> </ul> <p>When HCSPLTn.SplitEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transactions on transaction errors. This field must be set to at least 2'b01.</p> |       |     |
| 19:18 | <p>Endpoint Type (EPTyep)</p> <p>Indicates the transfer type selected.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: Control</li> <li>▪ 2'b01: Isochronous</li> <li>▪ 2'b10: Bulk</li> <li>▪ 2'b11: Interrupt</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 2'b0  | R_W |
| 17    | <p>Low-Speed Device (LSpdDev)</p> <p>This field is set by the application to indicate that this channel is communicating to a low-speed device.</p> <p>The application must program this bit when a low speed device is connected to the host through an FS HUB. The HS OTG Host core uses this field to drive the XCVR_SELECT signal to 2'b11 while communicating to the LS Device through the FS hub.</p> <p><b>Note:</b> In a peer to peer setup, the HS OTG Host core ignores this bit even if it is set by the application software.</p>                                                                                                                                              | 1'b0  | R_W |
| 16    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       | RO  |
| 15    | <p>Endpoint Direction (EPDir)</p> <p>Indicates whether the transaction is IN or OUT.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: OUT</li> <li>▪ 1'b1: IN</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 1'b0  | R_W |
| 14:11 | <p>Endpoint Number (EPNum)</p> <p>Indicates the endpoint number on the device serving as the data source or sink.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 4'h0  | R_W |
| 10:0  | <p>Maximum Packet Size (MPS)</p> <p>Indicates the maximum packet size of the associated endpoint.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 11'h0 | R_W |

### 33.5.4.9 Host Channel-n Split Control Register (HCSPLTn)

◇ Channel\_number:  $0 \leq n \leq 15$

◇ Offset:  $504h + (\text{Channel\_number} * 20h)$



**Table 33-36 Host Channel-n Split Control Register: HCSPLTn**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset | Acces |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31    | Split Enable (SpltEna)<br>The application sets this field to indicate that this channel is enabled to perform split transactions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 1'b0  | R_W   |
| 30:17 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |       | RO    |
| 16    | Do Complete Split (CompSplt)<br>The application sets this field to request the OTG host to perform a complete split transaction.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 1'b0  | R_W   |
| 15:14 | Transaction Position (XactPos)<br>This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. <ul style="list-style-type: none"> <li>▪ 2'b11: All. This is the entire data payload is of this transaction (which is less than or equal to 188 bytes).</li> <li>▪ 2'b10: Begin. This is the first data payload of this transaction (which is larger than 188 bytes).</li> <li>▪ 2'b00: Mid. This is the middle payload of this transaction (which is larger than 188 bytes).</li> <li>▪ 2'b01: End. This is the last payload of this transaction (which is larger than 188 bytes).</li> </ul> | 2'h0  | R_W   |
| 13:7  | Hub Address (HubAddr)<br>This field holds the device address of the transaction translator's hub.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 7'h0  | R_W   |
| 6:0   | Port Address (PrtAddr)<br>This field is the port number of the recipient transaction translator.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 7'h0  | R_W   |

#### 33.5.4.10 Host Channel-n Interrupt Register (HCINTn)

- ◇ Channel\_number:  $0 \leq n \leq 15$
- ◇ Offset:  $508h + (\text{Channel\_number} * 20h)$

This register indicates the status of a channel with respect to USB- and AHB-related events. It is shown in Figure 33-2. The application must read this register when the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt) is set. Before the application can read this register, it must first read the Host All Channels Interrupt (HAINT) register to get the exact channel number for the Host Channel-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the HAINT and GINTSTS registers.

**Table 33-37 Host Channel-n Interrupt Register: HCINTn**

| Field | Description | Reset | Acces |
|-------|-------------|-------|-------|
|-------|-------------|-------|-------|

|       |                                                                                                                                                                                                                                                                                                                                   |      |         |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------|
| 31:11 | Reserved                                                                                                                                                                                                                                                                                                                          |      | RO      |
| 13    | Descriptor rollover interrupt (DESC_LST_ROLLIntr)<br>This bit is valid only when Scatter/Gather DMA mode is enabled.<br>The core sets this bit when the corresponding channel's descriptor list rolls over.<br>For non Scatter/Gather DMA mode, this bit is reserved.                                                             | 1'b0 | R_SS_WC |
| 12    | Excessive Transaction Error (XCS_XACT_ERR)<br>This bit is valid only when Scatter/Gather DMA mode is enabled.<br>The core sets this bit when 3 consecutive transaction errors occurred on the USB bus. XCS_XACT_ERR will not be generated for Isochronous channels.<br>For non Scatter/Gather DMA mode, this bit is reserved.     | 1'b0 | R_SS_WC |
| 11    | BNA (Buffer Not Available) Interrupt (BNAIntr) This bit is valid only when Scatter/Gather DMA mode is enabled. The core generates this interrupt when the descriptor accessed is not ready for the Core to process. BNA will not be generated for Isochronous channels.<br>For non Scatter/Gather DMA mode, this bit is reserved. | 1'b0 | R_SS_WC |
| 10    | Data Toggle Error (DataTglErr)<br>In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.                                                                                                                                                                                                                | 1'b0 | R_SS_WC |
| 9     | Frame Overrun (FrmOvrn)<br>In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core                                                                                                                                                                                                                        | 1'b0 | R_SS_WC |
| 8     | Babble Error (BblErr)<br>In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.                                                                                                                                                                                                                         | 1'b0 | R_SS_WC |
| 7     | Transaction Error (XactErr)<br>Indicates one of the following errors occurred on the USB. <ul style="list-style-type: none"> <li>▪ CRC check failure</li> <li>▪ Timeout</li> <li>▪ Bit stuff error</li> <li>▪ False EOP</li> </ul> In Scatter/Gather DMA mode, the interrupt due to this bit is not set.                          | 1'b0 | R_SS_WC |
| 6     | NYET Response Received Interrupt (NYET)<br>In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.                                                                                                                                                                                                       | 1'b0 | R_SS_WC |
| 5     | ACK Response Received/Transmitted Interrupt (ACK)<br>In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.                                                                                                                                                                                             | 1'b0 | R_SS_WC |
| 4     | NAK Response Received Interrupt (NAK)<br>In Scatter/Gather DMA mode, the interrupt due to this bit is                                                                                                                                                                                                                             | 1'b0 | R_SS_WC |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |         |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------|
|   | masked in the core.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |         |
| 3 | <b>STALL Response Received Interrupt (STALL)</b><br>In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 1'b0 | R_SS_WC |
| 2 | <b>AHB Error (AHBErr)</b><br>This is generated only in DMA mode when there is an AHB error during AHB read/write.<br>The application can read the corresponding channel's DMA address register to get the error address.                                                                                                                                                                                                                                                                                                                                                                             | 1'b0 | R_SS_WC |
| 1 | <b>Channel Halted (ChHltd)</b><br>In non Scatter/Gather DMA mode, it indicates the transfer completed abnormally either because of any USB transaction error or in response to disable request by the application or because of a completed transfer.<br>In Scatter/Gather DMA mode, this indicates that transfer completed due to any of the following <ul style="list-style-type: none"> <li>▪ EOL being set in descriptor</li> <li>▪ AHB error</li> <li>▪ Excessive transaction errors</li> <li>▪ In response to disable request by the application</li> <li>▪ Babble</li> <li>▪ Stall</li> </ul> | 1'b0 | R_SS_WC |
| 0 | <b>Transfer Completed (XferCompl)</b><br>For Scatter/Gather DMA mode, it indicates that current descriptor processing got completed with IOC bit set in its descriptor.<br>In non Scatter/Gather DMA mode, it indicates that Transfer completed normally without any errors.                                                                                                                                                                                                                                                                                                                         | 1'b0 | R_SS_WC |

### 33.5.4.11 Host Channel-*n* Interrupt Mask Register (HCINTMSK<sub>n</sub>)

- ✧ Channel\_number:  $0 \leq n \leq 15$
- ✧ Offset:  $50Ch + (\text{Channel\_number} * 20h)$

This register reflects the mask for each channel status described in the previous section.

- ✧ Mask interrupt: 1'b0
- ✧ Unmask interrupt: 1'b1

**Table 33-38 Host Channel-*n* Interrupt Mask Register: HCINTMSK<sub>n</sub>**

| Field | Description                                                                                                                              | Reset | Access |
|-------|------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| 31:11 | Reserved                                                                                                                                 | 1'b0  | RO     |
| 13    | Descriptor rollover interrupt Mask register<br>(DESC_LST_ROLLIntrMsk)<br>This bit is valid only when Scatter/Gather DMA mode is enabled. | 1'b0  | R_W    |

|    |                                                                                                                                                                                            |      |     |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
|    | In non Scatter/Gather DMA mode, this bit is reserved.                                                                                                                                      |      |     |
| 12 | Reserved                                                                                                                                                                                   | 1'b0 | RO  |
| 11 | Reserved                                                                                                                                                                                   | 1'b0 | RO  |
| 11 | BNA (Buffer Not Available) Interrupt mask register (BNAIntrMsk)<br>This bit is valid only when Scatter/Gather DMA mode is enabled.<br>In non Scatter/Gather DMA mode, this bit is reserved | 1'b0 | R_W |
| 10 | Data Toggle Error Mask (DataTglErrMsk)<br>This bit is not applicable in Scatter/Gather DMA mode.                                                                                           | 1'b0 | RO  |
| 9  | Frame Overrun Mask (FrmOvrnMsk)<br>This bit is not applicable in Scatter/Gather DMA mode.                                                                                                  | 1'b0 | R_W |
| 8  | Babble Error Mask (BblErrMsk)<br>This bit is not applicable in Scatter/Gather DMA mode.                                                                                                    | 1'b0 | R_W |
| 7  | Transaction Error Mask (XactErrMsk)<br>This bit is not applicable in Scatter/Gather DMA mode                                                                                               | 1'b0 | R_W |
| 6  | NYET Response Received Interrupt Mask (NyetMsk)<br>This bit is not applicable in Scatter/Gather DMA mode.                                                                                  | 1'b0 | R_W |
| 5  | ACK Response Received/Transmitted Interrupt Mask (AckMsk)<br>This bit is not applicable in Scatter/Gather DMA mode.                                                                        | 1'b0 | R_W |
| 4  | NAK Response Received Interrupt Mask (NakMsk)<br>This bit is not applicable in Scatter/Gather DMA mode.                                                                                    | 1'b0 | R_W |
| 3  | STALL Response Received Interrupt Mask (StallMsk)<br>This bit is not applicable in Scatter/Gather DMA mode.                                                                                |      |     |
| 2  | AHB Error Mask (AHBErrMsk)<br>Note: This bit is only accessible when OTG_ARCHITECTURE = 2                                                                                                  |      |     |
| 1  | Channel Halted Mask (ChHltdMsk)                                                                                                                                                            | 1'b0 | R_W |
| 0  | Transfer Completed Mask (XferComplMsk)                                                                                                                                                     | 1'b0 | R_W |

### 33.5.4.12 Host Channel-*n* Transfer Size Register (HCTSIZn)

◇ Channel\_number:  $0 \leq n \leq 15$

◇ Offset:  $510h + (\text{Channel\_number} * 20h)$

In Scatter/Gather DMA mode, the HCTSIZn register is defined as described in the table below.

**Table 33-39 Host Channel-*n* Transfer Size Register: HCTSIZn**

| Field | Description                                                                                                                                                                                                                                                          | Reset | Acces |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 30:29 | PID (Pid)<br>The application programs this field with the type of PID to use for the initial transaction.<br>The host maintains this field for the rest of the transfer.<br><ul style="list-style-type: none"> <li>▪ 2'b00: DATA0</li> <li>▪ 2'b01: DATA2</li> </ul> | 2'b00 | R_W   |

|                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |     |
|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
|                                                                   | <ul style="list-style-type: none"> <li>▪ 2'b10: DATA1</li> <li>▪ 2'b11: MDATA (non-control)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |      |     |
| 28:19                                                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 1'b0 | RO  |
| 18:16                                                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 1'b0 | RO  |
| 15:8                                                              | <p>NTD (Number of Transfer Descriptors)<br/> <b>(Non Isochronous)</b><br/>                     This value is in terms of number of descriptors. Maximum number of descriptor that can be present in the list is 64. The values can be from 0 to 63.</p> <ul style="list-style-type: none"> <li>▪ 0 - 1 descriptor.</li> <li>▪ 63 - 64 descriptors</li> </ul> <p>This field indicates the total number of descriptors present in that list. The core will wrap around after servicing NTD number of descriptors for that list.</p> <p><b>(Isochronous)</b><br/>                     This field indicates the number of descriptors present in that list.µframe</p> <p>The possible values for FS are</p> <ul style="list-style-type: none"> <li>▪ 1 - 2 descriptors</li> <li>▪ 3 - 4 descriptors</li> <li>▪ 7 - 8 descriptors</li> <li>▪ 15 - 16 descriptors</li> <li>▪ 31 - 32 descriptors</li> <li>▪ 63 - 64 descriptors</li> </ul> <p>The possible values for HS are</p> <ul style="list-style-type: none"> <li>▪ 7 - 8 descriptors</li> <li>▪ 15 - 16 descriptors</li> <li>▪ 31 - 32 descriptors</li> <li>▪ 63 - 64 descriptors</li> <li>▪ 127 - 128 descriptors</li> <li>▪ 255 - 256 descriptors</li> </ul> | 8'h0 | R_W |
| 7:0                                                               | <p>SCHED_INFO (Schedule information)<br/>                     Every bit in this 8 bit register indicates scheduling for that microframe. Bit 0 indicates scheduling for 1<sup>st</sup> microframe and bit 7 indicates scheduling for 8<sup>th</sup> microframe in that frame. A value of 8'b11111111 indicates that the corresponding interrupt channel is scheduled to issue a token every microframe in that frame. A value of 8'b10101010 indicates that the corresponding interrupt channel is scheduled to issue a token every alternate microframe starting with second microframe. Note that this field is applicable only for periodic (Isochronous and Interrupt) channels.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 8'h0 | R_W |
| <b>In Non-Scatter/Gather mode, HCTSIZn is defined as follows:</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |     |

|       |                                                                                                                                                                                                                                                                                                                                                                                               |       |     |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|
| 31    | <p>Do Ping (DoPng)</p> <p>This bit is used only for OUT transfers. Setting this field to 1 directs the host to do PING protocol.</p> <p><b>Note:</b> Do not set this bit for IN transfers. If this bit is set for IN transfers it disables the</p>                                                                                                                                            | 1'b0  | R_W |
| 30:29 | <p>PID (Pid)</p> <p>The application programs this field with the type of PID to use for the initial transaction.</p> <p>The host maintains this field for the rest of the transfer.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: DATA0</li> <li>▪ 2'b01: DATA2</li> <li>▪ 2'b10: DATA1</li> <li>▪ 2'b11: MDATA (non-control)/SETUP (control)</li> </ul>                                | 2'b00 | R_W |
| 28:19 | <p>Packet Count (PktCnt)</p> <p>This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN). The host decrements this count on every successful transmission or reception of an OUT/IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion.</p>                                   | 10'h0 | R_W |
| 18:0  | <p>Transfer Size (XferSize)</p> <p>For an OUT, this field is the number of data bytes the host sends during the transfer.</p> <p>For an IN, this field is the buffer size that the application has Reserved for the transfer.</p> <p>The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions (periodic and non-periodic).</p> | 19'h0 | R_W |

### 33.5.4.13 Host Channel-n DMA Address Register (HCDMA<sub>n</sub>)

- ◇ Channel\_number:  $0 \leq n \leq 15$
- ◇ Offset:  $514h + (\text{Channel\_number} * 20h)$

This register is used by the OTG host in the internal DMA mode to maintain the current buffer pointer for IN/OUT transactions. The starting DMA address must be DWORD-aligned.

**Table 33-40 Host Channel-n DMA Address Register: HCDMA<sub>n</sub>**

| Field                  | Description                                                                                                                                                                                                                 | Reset                                    | Access |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|--------|
| <b>Buffer DMA Mode</b> |                                                                                                                                                                                                                             |                                          |        |
| 31:0                   | <p>DMA Address (DMAAddr)</p> <p>This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB</p> | "X" if not programmed as the register is | R_W    |

|                                                                                                                               | transaction.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | in<br>SPRAM |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------|-----|--------------|--------|-----|----|---|----|---|----|---|-----|----|-----|----|---------|--|-----|---|---|---|---|---|---|---|----|---|----|---|----|---|--|--|
| <b>Descriptor DMA Mode</b>                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| <b>Note:</b> For Scatter/Gather DMA mode, this address is the start of the page address where the descriptor list is located. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 31:N<br>(Isoc)                                                                                                                | DMA Address (DMAAddr)<br>Non-Isochronous:<br>This field holds the start address of the 512 bytes page. The first descriptor in the list should be located in this address. The first descriptor may be or may not be ready. The core starts processing the list from the CTD value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 23'h0       | R_W   |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 31:9<br>(Non Isoc)                                                                                                            | Isochronous:<br>This field holds the address of the 2*(nTD+1) bytes of locations in which the isochronous descriptors are present where N is based on nTD as per Table below                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
|                                                                                                                               | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">31:N</td> <td style="width: 33%;">N-1:3</td> <td style="width: 33%;">2:0</td> </tr> <tr> <td>Base Address</td> <td>Offset</td> <td>000</td> </tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 31:N        | N-1:3 | 2:0 | Base Address | Offset | 000 |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 31:N                                                                                                                          | N-1:3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2:0         |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| Base Address                                                                                                                  | Offset                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 000         |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
|                                                                                                                               | <table border="1" style="width: 50%; border-collapse: collapse; margin-right: 20px;"> <thead> <tr> <th colspan="2">HS ISOC</th> </tr> <tr> <th>nTD</th> <th>N</th> </tr> </thead> <tbody> <tr><td>7</td><td>6</td></tr> <tr><td>15</td><td>7</td></tr> <tr><td>31</td><td>8</td></tr> <tr><td>63</td><td>9</td></tr> <tr><td>127</td><td>10</td></tr> <tr><td>255</td><td>11</td></tr> </tbody> </table> <table border="1" style="width: 50%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">FS ISOC</th> </tr> <tr> <th>nTD</th> <th>N</th> </tr> </thead> <tbody> <tr><td>1</td><td>4</td></tr> <tr><td>3</td><td>5</td></tr> <tr><td>7</td><td>6</td></tr> <tr><td>15</td><td>7</td></tr> <tr><td>31</td><td>8</td></tr> <tr><td>63</td><td>9</td></tr> </tbody> </table> | HS ISOC     |       | nTD | N            | 7      | 6   | 15 | 7 | 31 | 8 | 63 | 9 | 127 | 10 | 255 | 11 | FS ISOC |  | nTD | N | 1 | 4 | 3 | 5 | 7 | 6 | 15 | 7 | 31 | 8 | 63 | 9 |  |  |
| HS ISOC                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| nTD                                                                                                                           | N                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 7                                                                                                                             | 6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 15                                                                                                                            | 7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 31                                                                                                                            | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 63                                                                                                                            | 9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 127                                                                                                                           | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 255                                                                                                                           | 11                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| FS ISOC                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| nTD                                                                                                                           | N                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 1                                                                                                                             | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 3                                                                                                                             | 5                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 7                                                                                                                             | 6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 15                                                                                                                            | 7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 31                                                                                                                            | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 63                                                                                                                            | 9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| N-1:3<br>(Isoc)                                                                                                               | Current Transfer Desc (CTD):<br><b>Non Isochronous:</b><br>This value is in terms of number of descriptors. The values can be from 0 to 63.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 6'h0        | R/W   |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |
| 8:3<br>(Non Isoc)                                                                                                             | 0 - 1 descriptor.<br>63- 64 descriptors.<br>This field indicates the current descriptor processed in the list. This field is updated both by application and the core. For example, if the application enables the channel after programming CTD=5, then the core will start processing the 6 <sup>th</sup> descriptor. The address is obtained by adding a value of (8bytes*5=) 40(decimal) to DMAAddr.<br><b>Isochronous:</b><br>CTD for isochronous is based on the current frame/μframe value. Need to be set to zero by application.                                                                                                                                                                                                                                               |             |       |     |              |        |     |    |   |    |   |    |   |     |    |     |    |         |  |     |   |   |   |   |   |   |   |    |   |    |   |    |   |  |  |

|     |          |      |    |
|-----|----------|------|----|
| 2:0 | Reserved | 3'h0 | RO |
|-----|----------|------|----|

#### 33.5.4.14 Host Channel-n DMA Buffer Address Register (HCDMABn)

- ◇ Channel\_number: 0 n 15
- ◇ Offset: 51Ch + (Channel\_number \* 20h)

This register is present only in case of Scatter/Gather DMA. It is implemented in RAM instead of flop-based implementation. This register holds the current buffer address.

**Table 33-41 Host Channel-n DMA Buffer Address Register: HCDMABn**

| Field | Description                                                                                                                                                                                                                          | Reset                                          | Acces |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|-------|
| 31:0  | Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved. | "X" if not programmed as the register in SPRAM | RO    |

#### 33.5.4.15 Host Frame List Base Address Register (HFLBAddr)

- ◇ Offset: 41Ch

This register is present only in case of Scatter/Gather DMA. It is implemented as flops. This register holds the starting address of the Frame list information.

**Table 33-42 Host Frame List Base Address Register: HFLBAddr**

| Field | Description                                                                                                | Reset | Acces |
|-------|------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:0  | The starting address of the Frame list. This register is used only for Isochronous and Interrupt Channels. | 32'h0 | R/W   |

### 33.5.5 Device Mode Registers

These registers are visible only in Device mode and must not be accessed in Host mode, as the results are unknown. Some of them affect all the endpoints uniformly, while others affect only a specific endpoint.

Device Mode registers fall into two categories:

#### Device Logical IN Endpoint-Specific Registers

It's instantiates one set of endpoint registers per logical endpoint. A logical endpoint is unidirectional: it can be either IN or OUT. To represent a bidirectional endpoint, two logical endpoints are required, one for the IN direction and the other for the OUT direction. This is also true for control endpoints.

The registers and register fields described in this section can pertain to IN or OUT endpoints, or both, or specific endpoint types as noted.



### 33.5.5.1 Device Configuration Register (DCFG)

◇ Offset: 800h

This register configures the core in Device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

**Table 33-43 Device Configuration Register: DCFG**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset | Access |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| 31:26 | Resume Validation Period (ResValid)<br>This field controls the period when the core resumes from a suspend. When this bit is set, the core counts for the ResValid number of clock cycles to detect a valid resume.<br>This field is effective only when DCFG.Ena32KHzSusp is set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 6'd2  | R_W    |
| 25:24 | Periodic Scheduling Interval (PerSchIntvl)<br>PerSchIntvl must be programmed only for Scatter/Gather DMA mode.<br><b>Description:</b> This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)frame.<br><ul style="list-style-type: none"> <li>▪ When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data.</li> <li>▪ When no periodic endpoints are active, then the internal DMA engine services non- periodic endpoints, ignoring this field.</li> <li>▪ After the specified time within a (micro)frame, the DMA switches to fetching for non- periodic endpoints.</li> <li>▪ 2'b00: 25% of (micro)frame.</li> <li>▪ 2'b01: 50% of (micro)frame.</li> <li>▪ 2'b10: 75% of (micro)frame.</li> <li>▪ 2'b11: Reserved.</li> </ul> | 2'b00 | R_W    |
| 23    | Enable Scatter/Gather DMA in Device mode (DescDMA).<br>When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation.<br><b>NOTE:</b> This bit must be modified only once after a reset.<br>The following combinations are available for programming:<br><ul style="list-style-type: none"> <li>▪ GAHBCFG.DMAEn=0,DCFG.DescDMA=0 =&gt; Slave mode</li> <li>▪ GAHBCFG.DMAEn=0,DCFG.DescDMA=1 =&gt; Invalid</li> <li>▪ GAHBCFG.DMAEn=1,DCFG.DescDMA=0 =&gt; Buffered DMA mode</li> <li>▪ GAHBCFG.DMAEn=1,DCFG.DescDMA=1 =&gt; Scatter/Gather DMA mode</li> </ul>                                                                                                                                                                                                                                                                                                       | 1'b0  | R_W    |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |      |     |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
| 22:18 | <p>IN Endpoint Mismatch Count (EPMisCnt)</p> <p>This field is valid only in shared FIFO operation.</p> <p>The application programs this field with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.</p> | 5'h8 | R_W |
| 17:13 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |      | RO  |
| 13    | <p>Enable Device OUT NAK (EnDevOutNak)</p> <p>This bit enables setting NAK for Bulk OUT endpoints after the transfer is completed when the core is operating in Device Descriptor DMA mode.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: The core does not set NAK after Bulk OUT transfer complete</li> <li>▪ 1'b1: The core sets NAK after Bulk OUT transfer complete</li> </ul> <p>This bit is valid only when OTG_EN_DESC_DMA == 1'b1</p>                            | 1'b0 | R_W |
| 12:11 | <p>Periodic Frame Interval (PerFrlnt)</p> <p>Indicates the time within a (micro)frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro)frame is complete.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: 80% of the (micro)frame interval</li> <li>▪ 2'b01: 85%</li> <li>▪ 2'b10: 90%</li> <li>▪ 2'b11: 95%</li> </ul>                                   | 2'h0 | R_W |
| 10:4  | <p>Device Address (DevAddr)</p> <p>The application must program this field after every SetAddress control command.</p>                                                                                                                                                                                                                                                                                                                                                         | 7'h0 | R_W |
| 3     | <p>Enable 32-KHz Suspend Mode (Ena32KHzS)</p> <p>When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend. This bit can only be set if USB 1.1 Full-Speed Serial Transceiver Interface has been selected.</p> <p>If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero.</p>                                           | 1'd0 | R_W |
| 2     | <p>Non-Zero-Length Status OUT Handshake (NZStsOUTHShk)</p> <p>The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage.</p> <ul style="list-style-type: none"> <li>▪ 1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet</li> </ul>                                                     | 1'b0 | R_W |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |     |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
|     | to the application. <ul style="list-style-type: none"> <li>1'b0: Send the received OUT packet to the application (zero-length or nonzero-length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                      |      |     |
| 1:0 | Device Speed (DevSpd)<br>Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. <ul style="list-style-type: none"> <li>2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz)</li> <li>2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz)</li> <li>2'b10: Low speed (USB 1.1 FS transceiver clock is 48 MHz)</li> <li>2'b11: Full speed (USB 1.1 FS transceiver clock is 48 MHz)</li> </ul> | 2'b0 | R_W |

### 33.5.5.2 Device Control Register (DCTL)

◇ Offset: 804h

**Table 33-44 Device Control Register: DCTL**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset | Access |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| 31:17 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |       | RO     |
| 17    | Enable Continue on BNA (EnContOnBNA)<br>This bit enables the DWC_otg core to continue on BNA for Bulk OUT endpoints. With this feature enabled, when a Bulk OUT endpoint receives a BNA interrupt the core starts processing the descriptor that caused the BNA interrupt after the endpoint re-enables the endpoint. <ul style="list-style-type: none"> <li>1'b0: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the DOEPDMA descriptor.</li> <li>1'b1: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the descriptor that received the BNA interrupt.</li> </ul> This bit is valid only when OTG_EN_DESC_DMA == 1'b1. It is a one-time programmable after reset bit like any other DCTL register bits. | 1'b0  | R_W    |
| 16    | Set NAK automatically on babble (NakOnBble). The core sets                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 1'b0  | R_W    |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |     |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
|       | NAK automatically for the endpoint on which babble is received.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |     |
| 15    | <p>Ignore frame number for isochronous endpoints (IgnrFrmNum)</p> <p>Slave Mode (GAHBCFG.DMAEn=0):<br/>This bit is not valid in Slave mode and should not be programmed to 1. Non- Scatter/Gather DMA mode (GAHBCFG.DMAEn=1, DCFG.DescDMA=0):<br/>This bit is not used when Threshold mode is enabled and should not be programmed to 1.<br/>In non-Scatter/Gather DMA mode, the application receives transfer complete interrupt after transfers for multiple (micro)frames are completed.</p> <ul style="list-style-type: none"> <li>▪ When Scatter/Gather DMA mode is disabled, this field is used by the application to enable periodic transfer interrupt. The application can program periodic endpoint transfers for multiple (micro)frames. <ul style="list-style-type: none"> <li>– 0: Periodic transfer interrupt feature is disabled; the application must program transfers for periodic endpoints every (micro)frame</li> <li>– 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. This field is also used by the application to enable periodic transfer interrupts.</li> </ul> </li> </ul> <p>Scatter/Gather DMA Mode (GAHBCFG.DMAEn=1, DCFG.DescDMA=1):<br/>This bit is not applicable to high-speed, high-bandwidth transfers and should not be programmed to 1.<br/>In addition, this bit is not used when Threshold mode is enabled and should not be programmed to 1.</p> <ul style="list-style-type: none"> <li>▪ 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. <ul style="list-style-type: none"> <li>– 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. When this bit is set, there must be only one packet per descriptor.</li> </ul> </li> </ul> | 1'b0 | R_W |
| 14:13 | <p>Global Multi Count (GMC)</p> <p>GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for non- periodic end</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 2'h0 | R_W |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |      |     |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
|     | points. <ul style="list-style-type: none"> <li>▪ 2'b00: Invalid.</li> <li>▪ 2'b01: 1 packet.</li> <li>▪ 2'b10: 2 packets.</li> <li>▪ 2'b11: 3 packets.</li> </ul> The value of this field automatically changes to 2'h1 when DCFG.DescDMA is set to 1.<br>When Scatter/Gather DMA mode is disabled, this field is reserved and reads 2'b00.                                                                                                                                               |      |     |
| 12  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      | RO  |
| 11  | Power-On Programming Done (PWROnPrgDone)<br>The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode.                                                                                                                                                                                                                                                                                                                        | 1'b0 | WO  |
| 10  | Clear Global OUT NAK (CGOUTNak)<br>A write to this field clears the Global OUT NAK.                                                                                                                                                                                                                                                                                                                                                                                                       | 1'b0 | WO  |
| 9   | Set Global OUT NAK (SGOUTNak)<br>A write to this field sets the Global OUT NAK.<br>The application uses this bit to send a NAK handshake on all OUT endpoints.<br>The application must set the this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared.                                                                                                                                                      | 1'b0 | WO  |
| 8   | Clear Global Non-periodic IN NAK (CGNPInNak)<br>A write to this field clears the Global Non-periodic IN NAK.                                                                                                                                                                                                                                                                                                                                                                              | 1'b0 | WO  |
| 7   | Set Global Non-periodic IN NAK (SGNPInNak)<br>A write to this field sets the Global Non-periodic IN NAK. The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core can also set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation.<br>The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared. | 1'b0 | WO  |
| 6:4 | Test Control (TstCtl) <ul style="list-style-type: none"> <li>▪ 3'b000: Test mode disabled</li> <li>▪ 3'b001: Test_J mode</li> <li>▪ 3'b010: Test_K mode</li> <li>▪ 3'b011: Test_SE0_NAK mode</li> <li>▪ 3'b100: Test_Packet mode</li> <li>▪ 3'b101: Test_Force_Enable</li> </ul> Others: Reserved                                                                                                                                                                                         | 3'h0 | R_W |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |      |     |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
| 3 | <p>Global OUT NAK Status (GOUTNakSts)</p> <ul style="list-style-type: none"> <li>▪ 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings.</li> <li>▪ 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 1'b0 | WO  |
| 2 | <p>Global Non-periodic IN NAK Status (GNPINNakSts)</p> <ul style="list-style-type: none"> <li>▪ 1'b0: A handshake is sent out based on the data availability in the transmit FIFO.</li> <li>▪ 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 1'b0 | WO  |
| 1 | <p>Soft Disconnect (SftDiscon)</p> <p>The application uses this bit to signal the DWC_otg core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit.</p> <p>The minimum duration for which the core must keep this bit set is specified in <a href="#">Table 1-53</a>.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration.</li> <li>▪ 1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host.</li> </ul> <p>Note: This bit can be also used for ULPI/FS Serial interfaces.</p> <p>Note: This bit is not impacted by a soft reset.</p> | 1'b1 | R_W |
| 0 | <p>Remote Wakeup Signaling (RmtWkUpSig)</p> <p>When the application sets this bit, the core initiates remote signaling to wake the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1-15 ms after setting it. If LPM is enabled and the core is in the L1 (Sleep) state, when the application sets this bit, the core initiates L1 remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Sleep state. As specified in the LPM specification, the hardware automatically clears this bit 50 <math>\mu</math>s (<math>T_{L1DevDrvResume}</math>) after being set by the application. The application must not set this bit when GLPMCFG</p>                                                                                                                                                                                                                                               | 1'b0 | R_W |

|  |                                                        |  |  |
|--|--------------------------------------------------------|--|--|
|  | bRemoteWake from the previous LPM transaction is zero. |  |  |
|--|--------------------------------------------------------|--|--|

Following table lists the minimum duration under various conditions for which the Soft Disconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect. To accommodate clock jitter, it is recommended that the application add some extra delay to the specified minimum duration.

**Table 33-45 Minimum Duration for Soft Disconnect**

| Operating Speed      | Device State                                       | Minimum Duration   |
|----------------------|----------------------------------------------------|--------------------|
| High speed           | Suspended                                          | 1 ms + 2.5 $\mu$ s |
| High speed           | Idle                                               | 3 ms + 2.5 $\mu$ s |
| High speed           | Not Idle or Suspended<br>(Performing transactions) | 125 $\mu$ s        |
| Full speed/Low speed | Suspended                                          | 1 ms + 2.5 $\mu$ s |
| Full speed/Low speed | Idle                                               | 2.5 $\mu$ s        |
| Full speed/Low speed | Not Idle or Suspended<br>(Performing transactions) | 2.5 $\mu$ s        |

### 33.5.5.3 Device Status Register (DSTS)

✧ Offset: 808h

This register indicates the status of the core with respect to USB-related events. It must be read on interrupts from Device All Interrupts (DAINT) register.

**Table 33-46 Device Status Register: DSTS**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset | Acces |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:22 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       | RO    |
| 21:8  | Frame or Microframe Number of the Received SOF (SOFFN)<br>When the core is operating at high speed, this field contains a microframe number.<br>When the core is operating at full or low speed, this field contains a frame number.<br><b>Note:</b> This register may return a non zero value if read immediately after power on reset.<br>In case the register bit reads non zero immediately after power on reset it does not indicate that SOF has been received from the host. The read value of this interrupt is valid only after a valid connection between host and device is established. | 14'h0 | RO    |
| 7:4   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       | RO    |
| 3     | Erratic Error (ErrticErr)<br>The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error) seen on the UTMI+.<br>Due to erratic errors, the DWC_otg core goes into Suspended state and an interrupt is generated to the application with Early                                                                                                                                                                                                                                                                 | 1'b0  | RO    |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       |    |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----|
|     | Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover.                                                                                                                                                                                                                                                                                                                                                                       |       |    |
| 2:1 | <p>Enumerated Speed (EnumSpd)</p> <p>Indicates the speed at which the DWC_otg core has come up after speed detection through a chirp sequence.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: High speed (PHY clock is running at 30 or 60 MHz)</li> <li>▪ 2'b01: Full speed (PHY clock is running at 30 or 60 MHz)</li> <li>▪ 2'b10: Low speed (PHY clock is running at 48 MHz, internal phy_clk at 6 MHz)</li> <li>▪ 2'b11: Full speed (PHY clock is running at 48 MHz)</li> </ul> <p>Low speed is not supported for devices using a UTMI+ PHY.</p> | 2'h01 | RO |
| 0   | <p>Suspend Status (SuspSts)</p> <p>In Device mode, this bit is set as long as a Suspend condition is detected on the USB.</p> <p>The core enters the Suspended state when there is no activity on the utmi_linestate signal for an extended period of time. The core comes out of the suspend:</p> <ul style="list-style-type: none"> <li>▪ When there is any activity on the utmi_linestate signal</li> <li>▪ When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig).</li> </ul>                 | 1'b0  | RO |

### 33.5.5.4 Device IN Endpoint Common Interrupt Mask Register (DIEPMSK)

◇ Offset: 810h

This register works with each of the Device IN Endpoint Interrupt (DIEPINTn) registers for all endpoints to generate an interrupt per IN endpoint. The IN endpoint interrupt for a specific status in the DIEPINTn register can be masked by writing to the corresponding bit in this register. Status bits are masked by default.

◇ Mask interrupt: 1'b0

◇ Unmask interrupt: 1'b1

**Table 33-47 Device IN Endpoint Common Interrupt Mask Register: DIEPMSK**

| Field | Description                                                                                      | Reset | Access |
|-------|--------------------------------------------------------------------------------------------------|-------|--------|
| 31:14 | Reserved                                                                                         |       | RO     |
| 13    | NAK interrupt Mask (NAKMsk)                                                                      | 1'h0  | R_W    |
| 12:10 | Reserved                                                                                         |       | RO     |
| 9     | BNA Interrupt Mask (BNAIntrMsk)<br>This bit is valid only when Device Descriptor DMA is enabled. | 1'b0  | R_W    |
| 8     | Fifo Underrun Mask (TxfifoUndrnMsk)                                                              | 1'b0  | R_W    |
| 7     | Reserved                                                                                         |       | RO     |
| 6     | IN Endpoint NAK Effective Mask (INEPNakEffMsk)                                                   | 1'b0  | R_W    |



|   |                                                                    |      |     |
|---|--------------------------------------------------------------------|------|-----|
| 5 | IN Token received with EP Mismatch Mask (INTknEPMisMsk)            | 1'b0 | R_W |
| 4 | IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk)          | 1'b0 | R_W |
| 3 | Timeout Condition Mask (TimeOUTMsk)<br>(Non-isochronous endpoints) | 1'b0 | R_W |
| 2 | AHB Error Mask (AHBErrMsk)                                         | 1'b0 | R_W |
| 1 | Endpoint Disabled Interrupt Mask (EPDisbldMsk)                     | 1'b0 | R_W |
| 0 | Transfer Completed Interrupt Mask (XferCompIMsk)                   | 1'b0 | R_W |

### 33.5.5.5 Device OUT Endpoint Common Interrupt Mask Register (DOEPMSK)

◇ Offset: 814h

This register works with each of the Device OUT Endpoint Interrupt (DOEPINTn) registers for all endpoints to generate an interrupt per OUT endpoint. The OUT endpoint interrupt for a specific status in the DOEPINTn register can be masked by writing into the corresponding bit in this register. Status bits are masked by default.

◇ Mask interrupt: 1'b0

◇ Unmask interrupt: 1'b1

**Table 33-48 Device OUT Endpoint Common Interrupt Mask Register: DOEPMSK**

| Field | Description                                                                                               | Reset | Acces |
|-------|-----------------------------------------------------------------------------------------------------------|-------|-------|
| 31:15 | Reserved                                                                                                  |       | RO    |
| 14    | NYET Interrupt Mask (NYETMsk)                                                                             | 1'h0  | R_W   |
| 13    | NAK Interrupt Mask (NAKMsk)                                                                               | 1'h0  | R_W   |
| 12    | Babble Interrupt Mask (BbleErrMsk)                                                                        | 1'h0  | R_W   |
| 11:10 | Reserved                                                                                                  |       | RO    |
| 9     | BNA interrupt Mask (BnaOutIntrMsk)                                                                        | 1'h0  | R_W   |
| 8     | OUT Packet Error Mask (OutPktErrMsk)                                                                      | 1'b0  | R_W   |
| 7     | Reserved                                                                                                  |       | RO    |
| 6     | Back-to-Back SETUP Packets Received Mask (Back2BackSETup)<br>Applies to control OUT endpoints only.       | 1'b0  | R_W   |
| 5     | Reserved                                                                                                  |       | RO    |
| 4     | OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk)<br>Applies to control OUT endpoints only. | 1'b0  | R_W   |
| 3     | SETUP Phase Done Mask (SetUPMsk)<br>Applies to control endpoints only.                                    | 1'b0  | R_W   |
| 2     | AHB Error (AHBErrMsk)                                                                                     | 1'b0  | R_W   |
| 1     | Endpoint Disabled Interrupt Mask (EPDisbldMsk)                                                            | 1'b0  | R_W   |
| 0     | Transfer Completed Interrupt Mask (XferCompIMsk)                                                          | 1'b0  | R_W   |

### 33.5.5.6 Device All Endpoints Interrupt Register (DAINT)

◇ Offset: 818h

When a significant event occurs on an endpoint, a Device All Endpoints Interrupt register interrupts the application using the Device OUT Endpoints Interrupt bit or Device IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively). This is shown in Figure 33-2. There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints.

For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Device Endpoint-n Interrupt register (DIEPINTn/DOEPINTn).

**Table 33-49 Device All Endpoints Interrupt Register: DAINT**

| Field | Description                                                                                                                  | Reset | Access |
|-------|------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| 31:16 | OUT Endpoint Interrupt Bits (OutEPInt)<br>One bit per OUT endpoint:<br>Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15 | 16'h0 | RO     |
| 15:0  | IN Endpoint Interrupt Bits (InEPInt)<br>One bit per IN Endpoint:<br>Bit 0 for IN endpoint 0, bit 15 for endpoint 15          | 16'h0 | RO     |

### 33.5.5.7 Device All Endpoints Interrupt Mask Register (DAINTMSK)

◇ Offset: 81Ch

The Device Endpoint Interrupt Mask register works with the Device Endpoint Interrupt register to interrupt the application when an event occurs on a device endpoint. However, the Device All Endpoints Interrupt (DAINT) register bit corresponding to that interrupt is still set.

◇ Mask Interrupt: 1'b0

◇ Unmask Interrupt: 1'b1

**Table 33-50 Device Endpoints Interrupt Mask Register: DAINTMSK**

| Field | Description                                                                                                                                                                                        | Reset | Access |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| 31:16 | OUT EP Interrupt Mask Bits (OutEpMsk)<br>One per OUT Endpoint:<br>Bit 16 for OUT EP 0, bit 31 for OUT EP 15<br>The value of this field depends on the number of OUT endpoints that are configured. | 16'h0 | R_W    |
| 15:0  | IN EP Interrupt Mask Bits (InEpMsk)<br>One bit per IN Endpoint:<br>Bit 0 for IN EP 0, bit 15 for IN EP 15<br>The value of this field depends on the number of IN endpoints                         | 16'h0 | R_W    |

|  |                      |  |  |
|--|----------------------|--|--|
|  | that are configured. |  |  |
|--|----------------------|--|--|

### 33.5.5.8 Device IN Token Sequence Learning Queue Read Register 1 (DTKNQR1)

✧ Offset: 820h

This register is valid only in non-periodic Shared FIFO operation (OTG\_EN\_DED\_TX\_FIFO = 0).

The queue is 4 bits wide to store the endpoint number. A read from this register returns the first 5 endpoint entries of the IN Token Sequence Learning Queue. When the queue is full, the new token is pushed into the queue and oldest token is discarded.

**Table 33-51 Device IN Token Sequence Learning Queue Read Register 1: DTKNQR1**

| Field | Description                                                                                                                                                                                                                                                                                                                                                      | Reset | Acces |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:8  | Endpoint Token (EPTkn)<br>Four bits per token represent the endpoint number of the token: <ul style="list-style-type: none"> <li>▪ Bits [31:28]: Endpoint number of Token 5</li> <li>▪ Bits [27:24]: Endpoint number of Token 4</li> <li>.....</li> <li>▪ Bits [15:12]: Endpoint number of Token 1</li> <li>▪ Bits [11:8]: Endpoint number of Token 0</li> </ul> | 24'h0 | RO    |
| 7     | Wrap Bit (WrapBit)<br>This bit is set when the write pointer wraps. It is cleared when the learning queue is cleared.                                                                                                                                                                                                                                            | 1'b0  | RO    |
| 6:5   | Reserved                                                                                                                                                                                                                                                                                                                                                         |       | RO    |
| 4:0   | IN Token Queue Write Pointer (INTknWPtr)                                                                                                                                                                                                                                                                                                                         | 5'h0  | RO    |

### 33.5.5.9 Device IN Token Sequence Learning Queue Read Register 2 (DTKNQR2)

✧ Offset: 0824h

This register is valid only in shared non-periodic Shared FIFO operation (OTG\_EN\_DED\_TX\_FIFO = 0).

A read from this register returns the next 8 endpoint entries of the learning queue.

**Table 33-52 Device IN Token Sequence Learning Queue Register 2: DTKNQR2**

| Field | Description                                                                                                                                                                                                                                                                                                                                                     | Reset | Acces |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:0  | Endpoint Token (EPTkn)<br>Four bits per token represent the endpoint number of the token: <ul style="list-style-type: none"> <li>▪ Bits [31:28]: Endpoint number of Token 13</li> <li>▪ Bits [27:24]: Endpoint number of Token 12</li> <li>.....</li> <li>▪ Bits [7:4]: Endpoint number of Token 7</li> <li>▪ Bits [3:0]: Endpoint number of Token 6</li> </ul> | 32'h0 | RO    |

### 33.5.5.10 Device IN Token Sequence Learning Queue Read Register 3 (DTKNQR3)

◇ Offset: 0830h

This register is valid only in non-periodic Shared FIFO operation (OTG\_EN\_DED\_TX\_FIFO = 0).

A read from this register returns the next 8 endpoint entries of the learning queue.

**Table 33-53 Device IN Token Sequence Learning Queue Register 3: DTKNQR3**

| Field | Description                                                                                                                                                                                                                                                                                                                                                       | Reset | Acces |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:0  | Endpoint Token (EPTkn)<br>Four bits per token represent the endpoint number of the token: <ul style="list-style-type: none"> <li>▪ Bits [31:28]: Endpoint number of Token 21</li> <li>▪ Bits [27:24]: Endpoint number of Token 20</li> <li>.....</li> <li>▪ Bits [7:4]: Endpoint number of Token 15</li> <li>▪ Bits [3:0]: Endpoint number of Token 14</li> </ul> | 32'h0 | RO    |

### 33.5.5.11 Device IN Token Sequence Learning Queue Read Register 4 (DTKNQR4)

◇ Offset: 0834h

This register is valid only in non-periodic Shared FIFO operation (OTG\_EN\_DED\_TX\_FIFO = 0).

A read from this register returns the last 8 endpoint entries of the learning queue.

**Table 33-54 Device IN Token Sequence Learning Queue Register 4: DTKNQR4**

| Field | Description                                                                                                                                                                                                                                                                                                                                                       | Reset | Acces |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:0  | Endpoint Token (EPTkn)<br>Four bits per token represent the endpoint number of the token: <ul style="list-style-type: none"> <li>▪ Bits [31:28]: Endpoint number of Token 29</li> <li>▪ Bits [27:24]: Endpoint number of Token 28</li> <li>.....</li> <li>▪ Bits [7:4]: Endpoint number of Token 23</li> <li>▪ Bits [3:0]: Endpoint number of Token 22</li> </ul> | 32'h0 | RO    |

### 33.5.5.12 Device VBUS Discharge Time Register (DVBUSDIS)

◇ Offset: 0828h

This register specifies the VBUS discharge time after VBUS pulsing during SRP.

**Table 33-55 Device VBUS Discharge Time Register: DVBUSDIS**

| Field | Description                                                                                                                  | Reset                                | Acces |
|-------|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|-------|
| 31:16 | Reserved                                                                                                                     |                                      | RO    |
| 15:0  | Device VBUS Discharge Time (DVBUSDis)<br>Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals: | 30 MHz: 16'h0B8F<br>60 MHz: 16'h17D7 | R_W   |

|  |                                                                                                                                                                                                                                 |  |  |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | VBUS discharge time in PHY clocks / 1,024<br>The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width).<br>Depending on your VBUS load, this value can need adjustment. |  |  |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

### 33.5.5.13 Device VBUS Pulsing Time Register (DVBUSPULSE)

◇ Offset: 082Ch

This register specifies the VBUS pulsing time during SRP.

**Table 33-56 Device VBUS Pulsing Time Register (DVBUSPULSE)**

| Field | Description                                                                                                                                                                                                                                                              | Reset                              | Acces |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|-------|
| 31:12 | Reserved                                                                                                                                                                                                                                                                 |                                    | RO    |
| 11:0  | Device VBUS Pulsing Time (DVBUSPulse)<br>Specifies the VBUS pulsing time during SRP. This value equals:<br>VBUS pulsing time in PHY clocks / 1,024<br>The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width). | 30 MHz: 12'h2C6<br>60 MHz: 12'h5B8 | R_W   |

### 33.5.5.14 Device Threshold Control Register (DTHRCTL)

◇ Offset: 830h

This register is valid only for device mode in Dedicated FIFO operation (OTG\_EN\_DED\_TX\_FIFO=1). Thresholding is not supported in Slave mode and so this register must not be programmed in Slave mode.

For threshold support, the AHB must be run at 60 MHz or higher.

**Table 33-57 Device Threshold Control Register (DTHRCTL)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                   | Reset | Acces |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:28 | Reserved                                                                                                                                                                                                                                                                                                                                                                                      |       | RO    |
| 27    | Arbiter Parking Enable (ArbPrkEn)<br>This bit controls internal DMA arbiter parking for IN endpoints. When thresholding is enabled and this bit is set to one, then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into underrun conditions. By default the parking is enabled.                                           | 1'b1  | R_W   |
| 26    | Reserved                                                                                                                                                                                                                                                                                                                                                                                      |       | RO    |
| 25:17 | Receive Threshold Length (RxThrLen)<br>This field specifies Receive thresholding size in DWORDS.<br>This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB.<br>The threshold length has to be at least eight DWORDS.<br>The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHB_CFG.HBstLen). | 9'h8  | R_W   |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |     |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
| 16    | <p>Receive Threshold Enable (RxThrEn)</p> <p>When this bit is set, the core enables thresholding in the receive direction.</p> <p>Note: Synopsys recommends that you do not enable RxThrEn, because it may cause issues in the RxFIFO especially during error conditions such as RxError and Babble.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                        | 1'b0 | R_W |
| 15:13 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |      | RO  |
| 12:11 | <p>AHB Threshold Ratio (AHBThrRatio)</p> <p>These bits define the ratio between the AHB threshold and the MAC threshold for the transmit path only. The AHB threshold always remains less than or equal to the USB threshold, because this does not increase overhead. Both the AHB and the MAC threshold must be DWORD-aligned. The application needs to program TxThrLen and the AHBThrRatio to make the AHB Threshold value DWORD aligned. If the AHB threshold value is not DWORD aligned, the core might not behave correctly. When programming the TxThrLen and AHBThrRatio, the application must ensure that the minimum AHB threshold value does not go below 8 DWORDS to meet the USB turnaround time requirements.</p>                | 2'h0 | R_W |
| 10:2  | <p>Transmit Threshold Length (TxThrLen)</p> <p>This field specifies Transmit thresholding size in DWORDS. This field also forms the MAC threshold and specifies the amount of data, in bytes, to be in the corresponding endpoint transmit FIFO before the core can start a transmit on the USB.</p> <p>When the value of AHBThrRatio is 2'h00, the threshold length must be at least 8 DWORDS. If the AHBThrRatio is nonzero, the application must ensure that the AHB threshold value does not go below the recommended 8 DWORDS.</p> <p>This field controls both isochronous and non-isochronous IN endpoint thresholds.</p> <p>The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).</p> | 9'h8 | R_W |
| 1     | <p>ISO IN Endpoints Threshold Enable. (ISOThrEn)</p> <p>When this bit is set, the core enables thresholding for isochronous IN endpoints.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 1'b0 | R_W |
| 0     | <p>Non-ISO IN Endpoints Threshold Enable. (NonISOThrEn)</p> <p>When this bit is set, the core enables thresholding for Non Isochronous IN endpoints.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 1'b0 | R_W |

### 33.5.5.15 Device IN Endpoint FIFO Empty Interrupt Mask Register: (DIEPEMPMSK)

✧ Offset: 834 h

This register is valid only in Dedicated FIFO operation (OTG\_EN\_DED\_TX\_FIFO = 1).

This register is used to control the IN endpoint FIFO empty interrupt generation (DIEPINTn.TxfEmp).

✧ Mask interrupt: 1'b0

◇ Unmask interrupt: 1'b1

**Table 33-58 Device IN Endpoint FIFO Empty Interrupt Mask Register: DIEPEMPSK**

| Field | Description                                                                                                                                                                                                                                                                   | Reset | Acces |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:16 | Reserved                                                                                                                                                                                                                                                                      |       | RO    |
| 15:0  | IN EP Tx FIFO Empty Interrupt Mask Bits (InEpTxfEmpMsk)<br>These bits acts as mask bits for DIEPINTn.<br>TxFEmp interrupt One bit per IN Endpoint: <ul style="list-style-type: none"> <li>▪ Bit 0 for IN endpoint 0</li> <li>...</li> <li>▪ Bit 15 for endpoint 15</li> </ul> | 16'h0 | R_W   |

### 33.5.5.16 Device Each Endpoint Interrupt Register (DEACHINT)

◇ Offset: 838h

Dependency: This register is available in device mode and only when parameter "Name: OTG\_MULTI\_PROC\_INTRPT".

There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints. For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used.

- ◇ Bits in this register are set and cleared when the application sets and clears bits in the corresponding Device Endpoint-n Interrupt register (DIEPINTn/DOEPINTn).
- ◇ The interrupt is automatically cleared once the DOEPINTn / DIEPINTn interrupt is cleared by the application.

**Table 33-59 Device Each Endpoint Interrupt Register: DEACHINT**

| Field | Description                                                                                                                                                                                             | Reset | Acces |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:16 | OUT Endpoint Interrupt Bits (EchOutEPInt)<br>One bit per OUT endpoint: <ul style="list-style-type: none"> <li>▪ Bit 16 for OUT endpoint 0</li> <li>...</li> <li>▪ Bit 31 for OUT endpoint 15</li> </ul> | 16'h0 | RO    |
| 15:0  | IN Endpoint Interrupt Bits (EchInEPInt)<br>One bit per IN Endpoint: <ul style="list-style-type: none"> <li>▪ Bit 0 for IN endpoint 0</li> <li>...</li> <li>▪ Bit 15 for IN endpoint 15</li> </ul>       | 16'h0 | RO    |

### 33.5.5.17 Device Each Endpoint Interrupt Register Mask (DEACHINTMSK)

◇ Offset: 83Ch

Dependency: This register is available only when parameter "Name: OTG\_MULTI\_PROC\_INTRPT".

The Device Each Endpoint Interrupt Mask register works with the Device Each Endpoint Interrupt register to interrupt the application when an event occurs on a device endpoint. However, the Device Each Endpoints Interrupt (DEACHINT) register bit corresponding to that interrupt remains set.

- ✧ Mask Interrupt: 1.b0
- ✧ Unmask Interrupt: 1.b1

**Table 33-60 Device Each Endpoint Interrupt Register Mask: DEACHINTMSK**

| Field | Description                                                                                                                                                                                         | Reset | Access |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| 31:16 | OUT EP Interrupt Mask Bits (EchOutEpMsk)<br>One per OUT Endpoint:<br><ul style="list-style-type: none"> <li>▪ Bit 16 for IN endpoint 0</li> <li>...</li> <li>▪ Bit 31 for IN endpoint 15</li> </ul> | 16'h0 | R_W    |
| 15:0  | IN EP Interrupt Mask Bits (EchInEpMsk)<br>One bit per IN Endpoint:<br><ul style="list-style-type: none"> <li>▪ Bit 0 for IN endpoint 0</li> <li>...</li> <li>▪ Bit 15 for IN endpoint 15</li> </ul> | 16'h0 | R_W    |

### 33.5.5.18 Device Each In Endpoint-*n* Interrupt Register (DIEPEACHMSK<sub>n</sub>)

- ✧ Offset 840h
- ✧ Endpoint\_number: 0= $\leq$  n  $\leq$  15
- ✧ Offset for IN endpoints: 840h + (Endpoint\_number \* 4h)

Dependency: This register is available in device mode and only when parameter "Name: OTG\_MULTI\_PROC\_INTRPT".

These registers are endpoint-specific mask registers for (DIEPINT<sub>n</sub>). The IN endpoint interrupt for a specific status in the DIEPINT<sub>n</sub> register can be masked by writing 0 to the corresponding bit in this register. Status bits are masked by default.

- ✧ Mask interrupt: 1'b0
- ✧ Unmask interrupt: 1'b1

**Table 33-61 Device Each In Endpoint-*n* Interrupt Register: DIEPEACHMSK<sub>n</sub>**

| Field | Description                                    | Reset | Access |
|-------|------------------------------------------------|-------|--------|
| 31:14 | Reserved                                       |       | RO     |
| 13    | NAK interrupt Mask (NAKMsk)                    | 1'b0  | R_W    |
| 12:10 | Reserved                                       |       | RO     |
| 9     | BNA interrupt Mask (BNAInIntrMsk)              | 1'b0  | R_W    |
| 8     | Fifo Under run Mask (TxfifoUndrnMsk)           | 1'b0  | R_W    |
| 7     | Reserved                                       |       | RO     |
| 6     | IN Endpoint NAK Effective Mask (INEPNakEffMsk) | 1'b0  | R_W    |



|   |                                                                 |      |     |
|---|-----------------------------------------------------------------|------|-----|
| 5 | IN Token received with EP Mismatch Mask (INTknEPMisMsk)         | 1'b0 | R_W |
| 4 | IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk)       | 1'b0 | R_W |
| 3 | Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints) | 1'b0 | R_W |
| 2 | AHB Error Mask (AHBErrMsk)                                      | 1'b0 | R_W |
| 1 | Endpoint Disabled Interrupt Mask (EPDisbldMsk)                  | 1'b0 | R_W |
| 0 | Transfer Completed Interrupt Mask (XferComplMsk)                | 1'b0 | R_W |

### 33.5.5.19 Device Each Out Endpoint-*n* Interrupt Register (DOEPEACHMSKn)

- ✧ Offset 880h
- ✧ Endpoint\_number:  $0 \leq n \leq 15$
- ✧ Offset for OUT endpoints:  $880h + (\text{Endpoint\_number} * 4h)$

Dependency: This register is available in device mode and only when parameter OTG\_MULTI\_PROC\_INTRPT=1.

These registers are endpoint specific mask registers for (DOEPINTn). The OUT endpoint interrupt for a specific status in the DOEPINTn register can be masked by writing 0 to the corresponding bit in this register. Status bits are masked by default.

- ✧ Mask interrupt: 1'b0
- ✧ Unmask interrupt: 1'b1

**Table 33-62 Device Each Out Endpoint-*n* Interrupt Register: DOEPEACHMSKn**

| Field | Description                                                                                            | Reset | Acces |
|-------|--------------------------------------------------------------------------------------------------------|-------|-------|
| 31:15 | Reserved                                                                                               |       | RO    |
| 14    | NYET interrupt Mask (NYETMsk)                                                                          | 1'b0  | R_W   |
| 13    | NAK interrupt Mask (NAKMsk)                                                                            | 1'b0  | R_W   |
| 12    | Babble interrupt Mask (BbleErrMsk)                                                                     | 1'b0  | R_W   |
| 11:10 | Reserved                                                                                               |       | RO    |
| 9     | BNA interrupt Mask (BnaOutIntrMsk)                                                                     | 1'b0  | R_W   |
| 8     | OUT Packet Error Mask (OutPktErrMsk)                                                                   | 1'b0  | R_W   |
| 7     | Reserved                                                                                               |       | RO    |
| 6     | Back-to-Back SETUP Packets Received Mask (Back2BackSETup)<br>Applies to control OUT endpoints only.    | 1'b0  | R_W   |
| 5     | Reserved                                                                                               |       | RO    |
| 4     | OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only. | 1'b0  | R_W   |
| 3     | SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only.                                    | 1'b0  | R_W   |
| 2     | AHB Error (AHBErrMsk)                                                                                  | 1'b0  | R_W   |

|   |                                                  |      |     |
|---|--------------------------------------------------|------|-----|
| 1 | Endpoint Disabled Interrupt Mask (EPDisbldMsk)   | 1'b0 | R_W |
| 0 | Transfer Completed Interrupt Mask (XferComplMsk) | 1'b0 | R_W |

### 33.5.5.20 Device Control IN Endpoint 0 Control Register (DIEPCTL0)

◇ Offset: 900h

This section describes the Control IN Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

**Table 33-63 Device Control IN Endpoint 0 Control Register: DIEPCTL0**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset | Access  |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|---------|
| 31    | <p>Endpoint Enable (EPEna)</p> <ul style="list-style-type: none"> <li>▪ When Scatter/Gather DMA mode is enabled, for IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup.</li> <li>▪ When Scatter/Gather DMA mode is disabled—such as in buffer-pointer based DMA mode—this bit indicates that data is ready to be transmitted on the endpoint.</li> </ul> <p>The core clears this bit before setting the following interrupts on this endpoint:</p> <ul style="list-style-type: none"> <li>▪ Endpoint Disabled</li> <li>▪ Transfer Completed</li> </ul> | 1'b0  | R_WS_SC |
| 30    | <p>Endpoint Disable (EPDis)</p> <p>The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.</p> <p>This bit is valid only when DMA mode is enabled.</p>                                                                                                             | 1'b0  | R_WS_SC |
| 29:28 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |       | RO      |
| 27    | <p>Set NAK (SNAK)</p> <p>A write to this bit sets the NAK bit for the endpoint.</p> <p>Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.</p>                                                                                                                                                                                                                                                                                                                             | 1'b0  | WO      |
| 26    | <p>Clear NAK (CNAK)</p> <p>A write to this bit clears the NAK bit for the endpoint.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 1'b0  | WO      |
| 25:22 | <p>TxFIFO Number (TxFNum)</p> <p>For Shared FIFO operation, this value is always set to 0, indicating that control IN endpoint 0 data is always written in the Non-Periodic Transmit FIFO.</p> <p>For Dedicated FIFO operation, this value is set to the FIFO number that is</p>                                                                                                                                                                                                                                                                                                                                        | 4'h0  | R_W     |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |      |         |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------|
|       | assigned to IN Endpoint 0.                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |         |
| 21    | <b>STALL Handshake (Stall)</b><br>The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.                                                                                                                                                                                       | 1'b0 | R_WS_SC |
| 20    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                       |      | RO      |
| 19:18 | <b>Endpoint Type (EPTypE)</b><br>Hardcoded to 00 for control.                                                                                                                                                                                                                                                                                                                                                                                                  | 2'h0 | RO      |
| 17    | <b>NAK Status (NAKSts)</b><br>Indicates the following:<br>1'b0: The core is transmitting non-NAK handshakes based on the FIFO status<br>1'b1: The core is transmitting NAK handshakes on this endpoint.<br>When this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. | 1'b0 | RO      |
| 16    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                       |      | RO      |
| 15    | <b>USB Active Endpoint (USBActEP)</b><br>This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.                                                                                                                                                                                                                                                                                                | 1'b1 | RO      |
| 14:11 | <b>Next Endpoint (NextEp)</b><br>Applies to non-periodic IN endpoints only.<br>Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is not set. This field is not valid in Slave mode.<br><b>Note:</b> This field is valid only for Shared FIFO operations.                                                                               | 4'b0 | R_W     |
| 10:2  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                       |      | RO      |
| 1:0   | <b>Maximum Packet Size (MPS)</b><br>Applies to IN and OUT endpoints.<br>The application must program this field with the maximum packet size for the current logical endpoint.                                                                                                                                                                                                                                                                                 | 2'h0 | R_W     |

### 33.5.5.21 Device Control OUT Endpoint 0 Control Register (DOEPTL0)

✧ Offset: B00h

This section describes the Control OUT Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

Table 33-64 Device OUT Endpoint 0 Control Register: DOEPTL0

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset | Access  |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|---------|
| 31    | <p>Endpoint Enable (EPEna)</p> <p>When Scatter/Gather DMA mode is enabled, for OUT endpoints this bit indicates that the descriptor structure and data buffer to receive data is setup.</p> <ul style="list-style-type: none"> <li>When Scatter/Gather DMA mode is disabled—(such as for buffer-pointer based DMA mode)—this bit indicates that the application has allocated the memory to start receiving data from the USB.</li> </ul> <p>The core clears this bit before setting any of the following interrupts on this endpoint:</p> <ul style="list-style-type: none"> <li>SETUP Phase Done</li> <li>Endpoint Disabled</li> <li>Transfer Completed</li> </ul> <p><b>Note:</b> In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory.</p> | 1'b0  | R_WS_SC |
| 30    | <p>Endpoint Disable (EPDis)</p> <p>The application cannot disable control OUT endpoint 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 1'b0  | RO      |
| 29:28 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       | RO      |
| 27    | <p>Set NAK (SNAK)</p> <p>A write to this bit sets the NAK bit for the endpoint.</p> <p>Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 1'b0  | WO      |
| 26    | <p>Clear NAK (CNAK)</p> <p>A write to this bit clears the NAK bit for the endpoint.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 1'b0  | WO      |
| 25:22 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |       | RO      |
| 21    | <p>STALL Handshake (Stall)</p> <p>The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.</p>                                                                                                                                                                                                                                                                                                                                                                                                                           | 1'b0  | R_WS_SC |
| 20    | <p>Snoop Mode (Snp)</p> <p>This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 1'b0  | R_W     |
| 19:18 | <p>Endpoint Type (EPTyp)</p> <p>Hardcoded to 2'b00 for control.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 2'h0  | RO      |
| 17    | <p>NAK Status (NAKSts)</p> <p>Indicates the following:</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1'b0  | RO      |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                               |      |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|----|
|      | 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.<br>1'b1: The core is transmitting NAK handshakes on this endpoint.<br>When either the application or the core sets this bit, the core stops receiving data, even if there is space in the RxFIFO to accommodate the incoming packet.<br>Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake. |      |    |
| 16   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                      |      | RO |
| 15   | USB Active Endpoint (USBActEP)<br>This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.                                                                                                                                                                                                                                                                    | 1'b1 | RO |
| 14:2 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                      |      | RO |
| 1:0  | Maximum Packet Size (MPS)<br>The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. <ul style="list-style-type: none"> <li>▪ 2'b00: 64 bytes</li> <li>▪ 2'b01: 32 bytes</li> <li>▪ 2'b10: 16 bytes</li> <li>▪ 2'b11: 8 bytes</li> </ul>                                                                                                                               | 2'h0 | RO |

### 33.5.5.22 Device Endpoint-*n* Control Register (DIEPCTL<sub>*n*</sub>/DOEPC<sub>*n*</sub>)

- ◇ Endpoint\_number:  $1 \leq n \leq 15$
- ◇ Offset for IN endpoints:  $900h + (\text{Endpoint\_number} * 20h)$
- ◇ Offset for OUT endpoints:  $B00h + (\text{Endpoint\_number} * 20h)$

The application uses this register to control the behavior of each logical endpoint other than endpoint 0.

**Table 33-65 Device Endpoint-*n* Control Register: DIEPCTL<sub>*n*</sub>/DOEPC<sub>*n*</sub>**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset | Acces   |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|---------|
| 31    | Endpoint Enable (EPEna)<br>Applies to IN and OUT endpoints. <ul style="list-style-type: none"> <li>▪ When Scatter/Gather DMA mode is enabled,</li> <li>▪ For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup.</li> <li>▪ For OUT endpoint it indicates that the descriptor structure and data buffer to</li> <li>▪ receive data is setup.</li> <li>▪ When Scatter/Gather DMA mode is enabled—such as for buffer-pointer based DMA mode:                             <ul style="list-style-type: none"> <li>– For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint.</li> </ul> </li> </ul> | 1'b0  | R_WS_SC |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |              |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|--------------|
|    | <ul style="list-style-type: none"> <li>- For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB.</li> <li>- The core clears this bit before setting any of the following interrupts on this endpoint. <ul style="list-style-type: none"> <li>▪ SETUP Phase Done</li> <li>▪ Endpoint Disabled</li> <li>▪ Transfer Completed</li> </ul> </li> </ul> <p><b>Note:</b> For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.</p>                                                                                                                                                                                                            |      |              |
| 30 | <p>Endpoint Disable (EPDis)</p> <p>Applies to IN and OUT endpoints.</p> <p>The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.</p>                                                                                                                                                                                                                                                           | 1'b0 | R_WS_SC      |
| 29 | <p>Set DATA1 PID (SetD1PID)</p> <p>Applies to interrupt/bulk IN and OUT endpoints only.</p> <p>Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.</p> <p>This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.</p> <p>Set Odd (micro)frame (SetOddFr)</p> <p>Applies to isochronous IN and OUT endpoints only.</p> <p>Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to odd (micro)frame.</p> <p>This field is not applicable for Scatter/Gather DMA mode.</p>                                                                                                                                                                                                | 1'b0 | WO<br><br>WO |
| 28 | <p>Set DATA0 PID (SetD0PID)</p> <p>Applies to interrupt/bulk IN and OUT endpoints only.</p> <p>Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0.</p> <p>This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.</p> <p>In non-Scatter/Gather DMA mode: Set Even (micro)frame (SetEvenFr)</p> <p>Applies to isochronous IN and OUT endpoints only.</p> <p>Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to even (micro) frame.</p> <p>When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is in the transmit descriptor structure. The frame in which to receive data is updated in receive descriptor</p> | 1'b0 | WO           |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |      |                    |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|--------------------|
|       | structure.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |                    |
| 27    | <p>Set NAK (SNAK)</p> <p>Applies to IN and OUT endpoints.</p> <p>A write to this bit sets the NAK bit for the endpoint.</p> <p>Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 1'b0 | WO                 |
| 26    | <p>Clear NAK (CNAK)</p> <p>Applies to IN and OUT endpoints.</p> <p>A write to this bit clears the NAK bit for the endpoint.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 1'b0 | WO                 |
| 25:22 | <p>TxFIFO Number (TxFNum)</p> <p><b>Shared FIFO Operation</b>—non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic TxFIFO number.</p> <ul style="list-style-type: none"> <li>▪ 4'h0: Non-Periodic TxFIFO</li> <li>▪ Others: Specified Periodic TxFIFO.number</li> </ul> <p><b>Note:</b> An interrupt IN endpoint can be configured as a non-periodic endpoint for applications such as mass storage. The core treats an IN endpoint as a non-periodic endpoint if the TxFNum field is set to 0. Otherwise, a separate periodic FIFO must be allocated for an interrupt IN endpoint, and the number of this FIFO must be programmed into the TxFNum field. Configuring an interrupt IN endpoint as a non-periodic endpoint saves the extra periodic FIFO area.</p> <p><b>Dedicated FIFO Operation</b>—these bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number.</p> <p>This field is valid only for IN endpoints.</p> | 4'h0 | R_W                |
| 21    | <p>STALL Handshake (Stall)</p> <p>Applies to non-control, non-isochronous IN and OUT endpoints only.</p> <p>The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core.</p> <p>Applies to control endpoints only.</p> <p>The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.</p>                                                                                                                                                                                                                                                                         | 1'h0 | R_W<br><br>R_WS_SC |
| 20    | Snoop Mode (Snp)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 1'b0 | R_W                |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |     |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|
|       | <p>Applies to OUT endpoints only.</p> <p>This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |     |
| 19:18 | <p>Endpoint Type (EPTYPE)</p> <p>Applies to IN and OUT endpoints.</p> <p>This is the transfer type supported by this logical endpoint.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: Control</li> <li>▪ 2'b01: Isochronous</li> <li>▪ 2'b10: Bulk</li> <li>▪ 2'b11: Interrupt</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 2'h0 | R_W |
| 17    | <p>NAK Status (NAKSts)</p> <p>Applies to IN and OUT endpoints.</p> <p>Indicates the following:</p> <ul style="list-style-type: none"> <li>▪ 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status.</li> <li>▪ 1'b1: The core is transmitting NAK handshakes on this endpoint.</li> </ul> <p>When either the application or the core sets this bit:</p> <p>The core stops receiving any data on an OUT endpoint, even if there is space in the Rx FIFO to accommodate the incoming packet.</p> <p>For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the Tx FIFO.</p> <p>For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the Tx FIFO.</p> <p>Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.</p>             | 1'b0 | RO  |
| 16    | <p>Endpoint Data PID (DPID)</p> <p>Applies to interrupt/bulk IN and OUT endpoints only.</p> <p>Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated.</p> <p>The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID.</p> <ul style="list-style-type: none"> <li>▪ 1'b0: DATA0</li> <li>▪ 1'b1: DATA1</li> </ul> <p>This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.</p> <p>Even/Odd (Micro)Frame (EO_FrNum)</p> <p>In non-Scatter/Gather DMA mode:</p> <p>Applies to isochronous IN and OUT endpoints only.</p> <p>Indicates the (micro)frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the</p> | 1'b0 | RO  |



|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |       |        |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
|       | even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register. <ul style="list-style-type: none"> <li>▪ 1'b0: Even (micro)frame</li> <li>▪ 1'b1: Odd (micro)frame</li> </ul> When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is provided in the transmit descriptor structure. The frame in which data is received is updated in receive descriptor structure. |       |        |
| 15    | USB Active Endpoint (USBActEP)<br>Applies to IN and OUT endpoints.<br>Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.                                                                                                                       | 1'b0  | R_W_SC |
| 14:11 | Next Endpoint (NextEp)<br>Applies to non-periodic IN endpoints only.<br>Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is low. This field is not valid in Slave mode operation.<br><b>Note:</b> This field is valid only for Shared FIFO operations.                                                                                                                                           | 4'b0  | R_W    |
| 10:0  | Maximum Packet Size (MPS)<br>Applies to IN and OUT endpoints.<br>The application must program this field with the maximum packet size for the current                                                                                                                                                                                                                                                                                                                                                                     | 11'h0 | R_W    |

### 33.5.5.23 Device Endpoint-*n* Interrupt Register (DIEPINT<sub>*n*</sub>/DOEPINT<sub>*n*</sub>)

- ◇ Endpoint\_number:  $0 \leq n \leq 15$
- ◇ Offset for IN endpoints:  $908h + (\text{Endpoint\_number} * 20h)$
- ◇ Offset for OUT endpoints:  $B08h + (\text{Endpoint\_number} * 20h)$

This register indicates the status of an endpoint with respect to USB- and AHB-related events. It is shown in Figure 33-2. The application must read this register when the OUT Endpoints Interrupt bit or IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively) is set. Before the application can read this register, it must first read the Device All Endpoints Interrupt (DAINT) register to get the exact endpoint number for the Device Endpoint-*n* Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the DAINTE and GINTSTS registers.

**Table 33-66 Device Endpoint-*n* Interrupt Register: DIEPINT<sub>*n*</sub>/DOEPINT<sub>*n*</sub>**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset | Acces                  |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------------------|
| 31:15 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       | RO                     |
| 14    | <p>NYET interrupt (NYETIntrpt)</p> <p>The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 1'b0  | R_SS_WC                |
| 13    | <p>NAK interrupt (NAKIntrpt)</p> <p>The core generates this interrupt when a NAK is transmitted. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to unavailability of data in the TXFifo.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1'b0  | R_SS_WC                |
| 12    | <p>BbleErr (Babble Error) interrupt (BbleErrIntrpt)</p> <p>The core generates this interrupt when babble is received for the endpoint.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 1'b0  | R_SS_WC                |
| 11    | <p>PktDrpSts (Packet Dropped Status)</p> <p>This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt.</p> <p><b>Dependency:</b> This bit is valid in non Scatter/Gather DMA mode when periodic transfer interrupt feature is selected.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 1'b0  | R_SS_WC                |
| 10    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       | RO                     |
| 9     | <p>BNA (Buffer Not Available) Interrupt (BNAIntr)</p> <p>The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done.</p> <p><b>Dependency:</b> This bit is valid only when Scatter/Gather DMA mode is enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 1'b0  | R_SS_WC                |
| 8     | <p>FIFO Underrun (TxfifoUndrn)</p> <p>Applies to IN endpoints only</p> <p>The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint.</p> <p><b>Dependency:</b> This interrupt is valid only when both of the following conditions are true:</p> <ul style="list-style-type: none"> <li>▪ Parameter OTG_EN_DED_TX_FIFO=1</li> <li>▪ Thresholding is enabled</li> </ul> <p>OUT Packet Error (OutPktErr)</p> <p>Applies to OUT endpoints only This interrupt is asserted when the core detects an overflow or a CRC error for an OUT packet.</p> <p><b>Dependency:</b> This interrupt is valid only when both of the following conditions are true:</p> <ul style="list-style-type: none"> <li>▪ Parameter OTG_EN_DED_TX_FIFO=1</li> <li>▪ Thresholding is enabled.</li> </ul> | 1'b0  | R_SS_WC<br><br>R_SS_WC |
| 7     | <p>Transmit FIFO Empty (TxFEmp)</p> <p>This bit is valid only for IN Endpoints</p> <p>This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1'b1  | RO                     |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |         |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------|
|   | the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |         |
| 6 | <p>N Endpoint NAK Effective (INEPNakEff)</p> <p>This bit should be cleared by writing a 1'b1 before writing a 1'b1 to corresponding DIEPCTLn.CNAK.</p> <p>The interrupt indicates that the IN endpoint NAK bit set by the application has taken effect in the core.</p> <p>This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit.</p> <p>This bit is applicable only when the endpoint is enabled.</p>                                                                                                                                                                                                       | 1'b0 | R_SS_WC |
|   | <p>Back-to-Back SETUP Packets Received (Back2BackSETup)</p> <p>Applies to Control OUT endpoints only.</p> <p>This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint.</p> <p>This bit is not valid in Slave mode.</p>                                                                                                                                                                                                                                                                                                                                                                                         | 1'b0 | R_SS_WC |
| 5 | <p>IN Token Received with EP Mismatch (INTknEPMis)</p> <p>Applies to non-periodic IN endpoints only.</p> <p>Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received.</p>                                                                                                                                                                                                                                                                                                                                | 1'b0 | R_SS_WC |
|   | <p>Status Phase Received For Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints and only in Scatter Gather DMA mode.</p> <p>This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer.</p> <p>The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in case of Scatter Gather DMA mode.</p> |      |         |
| 4 | <p>IN Token Received When TxFIFO is Empty (INTknTXFEmp)</p> <p>Indicates that an IN token was received when the associated TxFIFO (periodic/non- periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received.</p>                                                                                                                                                                                                                                                                                                                                                                                                                   | 1'b0 | R_SS_WC |
|   | <p>OUT Token Received When Endpoint Disabled (OUTTknEPdis)</p> <p>Indicates that an OUT token was received when the endpoint was not yet enabled.</p> <p>This interrupt is asserted on the endpoint for which the OUT token was received.</p>                                                                                                                                                                                                                                                                                                                                                                                                                               |      |         |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |         |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------|
| 3 | <p>Timeout Condition (TimeOUT)</p> <ul style="list-style-type: none"> <li>▪ In shared TX FIFO mode, applies to non-isochronous IN endpoints only.</li> <li>▪ In dedicated FIFO mode, applies only to Control IN endpoints.</li> <li>▪ In Scatter/Gather DMA mode, the TimeOUT interrupt is not asserted.</li> </ul> <p>Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint.</p> <p>SETUP Phase Done (SetUp)</p> <p>Applies to control OUT endpoints only.</p> <p>Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.</p>                               | 1'b0 | R_SS_WC |
| 2 | <p>AHB Error (AHBErr)</p> <p>Applies to IN and OUT endpoints.</p> <p>This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 1'b0 | R_SS_WC |
| 1 | <p>Endpoint Disabled Interrupt (EPDisbld)</p> <p>Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 1'b0 | R_SS_WC |
| 0 | <p>Transfer Completed Interrupt (XferCompl)</p> <p>Applies to IN and OUT endpoints.</p> <ul style="list-style-type: none"> <li>▪ When Scatter/Gather DMA mode is enabled</li> <li>▪ For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO.</li> <li>▪ For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set.</li> <li>▪ When Scatter/Gather DMA mode is disabled, this field indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.</li> </ul> | 1'b0 | R_SS_WC |

### 33.5.5.24 Device Endpoint 0 Transfer Size Register (DIEPTSIZ0/DOEPTSIZ0)

✧ Offset for IN endpoints: 910h

✧ Offset for OUT endpoints: B10h

The application must modify this register before enabling endpoint 0. Once endpoint 0 is enabled using Endpoint Enable bit of the Device Control Endpoint 0 Control registers (DIEPCTL0.EPEna/DOEPCTL0.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit.

Nonzero endpoints use the registers for endpoints 1-15.

When Scatter/Gather DMA mode is enabled, this register must not be programmed by the application. If the application reads this register when Scatter/Gather DMA mode is enabled, the core returns all zeros.

**Table 33-67 Device IN Endpoint 0 Transfer Size Register: DIEPTSIZ0**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                          | Reset | Acces |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31:21 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                             |       | RO    |
| 20:19 | Packet Count (PktCnt)<br>Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0.<br>This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO.                                                                                                                                                                         | 2'b0  | R_W   |
| 18:7  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                             |       | RO    |
| 6:0   | Transfer Size (XferSize)<br>Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data.<br>The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet.<br>The core decrements this field every time a packet from the external memory is written to the TxFIFO. | 7'h0  | R_W   |

**Table 33-68 Device OUT Endpoint 0 Transfer Size Register: DOEPTSIZ0**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                        | Reset | Acces |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| 31    | Reserved                                                                                                                                                                                                                                                                                                                                                                           |       | RO    |
| 30:29 | SETUP Packet Count (SUPCnt)<br>This field specifies the number of back-to-back SETUP data packets the endpoint can receive. <ul style="list-style-type: none"> <li>▪ 2'b01: 1 packet</li> <li>▪ 2'b10: 2 packets</li> <li>▪ 2'b11: 3 packets</li> </ul>                                                                                                                            | 2'h0  | R_W   |
| 28:20 | Reserved                                                                                                                                                                                                                                                                                                                                                                           |       | RO    |
| 19    | Packet Count (PktCnt) This field is decremented to zero after a packet is written into the RxFIFO.                                                                                                                                                                                                                                                                                 | 1'b0  | R_W   |
| 18:7  | Reserved                                                                                                                                                                                                                                                                                                                                                                           |       | RO    |
| 6:0   | Transfer Size (XferSize)<br>Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data.<br>The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet.<br>The core decrements this field every time a packet is read from the | 7'h0  | R_W   |

|  |                                            |  |  |
|--|--------------------------------------------|--|--|
|  | RxFIFO and written to the external memory. |  |  |
|--|--------------------------------------------|--|--|

### 33.5.5.25 Device Endpoint-*n* Transfer Size Register (DIEPTSIZn/DOEPTSIZn)

- ✧ Endpoint\_number:  $1 \leq n \leq 15$
- ✧ Offset for IN endpoints:  $910h + (\text{Endpoint\_number} * 20h)$
- ✧ Offset for OUT endpoints:  $B10h + (\text{Endpoint\_number} * 20h)$

The application must modify this register before enabling the endpoint. Once the endpoint is enabled using Endpoint Enable bit of the Device Endpoint-*n* Control registers (DIEPCTLn.EPEna/DOEPCTLn.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit.

This register is used only for endpoints other than Endpoint 0.

**Table 33-69 Device Endpoint-*n* Transfer Size Register: DIEPTSIZn/DOEPTSIZn**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset | Access                                    |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------------------------------------------|
| 31    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |       | RO                                        |
| 30:29 | <p>Multi Count (MC)<br/>Applies to IN endpoints only.<br/>For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints.</p> <ul style="list-style-type: none"> <li>▪ 2'b01: 1 packet</li> <li>▪ 2'b10: 2 packets</li> <li>▪ 2'b11: 3 packets</li> </ul> <p>For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-<i>n</i> Control register (DIEPCTLn.NextEp).</p> <p>Received Data PID (RxDPID)<br/>Applies to isochronous OUT endpoints only.<br/>This is the data PID received in the last packet for this endpoint.</p> <ul style="list-style-type: none"> <li>▪ 2'b00: DATA0</li> <li>▪ 2'b01: DATA2</li> <li>▪ 2'b10: DATA1</li> <li>▪ 2'b11: MDATA</li> </ul> <p>SETUP Packet Count (SUPCnt)<br/>Applies to control OUT Endpoints only.<br/>This field specifies the number of back-to-back SETUP data packets the endpoint can receive.</p> | 2'b0  | <p>R_W</p> <p>RO</p> <p>RO</p> <p>R_W</p> |

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |       |     |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-----|
|       | <ul style="list-style-type: none"> <li>▪ 2'b01: 1 packet</li> <li>▪ 2'b10: 2 packets</li> <li>▪ 2'b11: 3 packets</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |       |     |
| 28:19 | <p>Packet Count (PktCnt)</p> <p>Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint (<math>PktCnt = XferSize / MPS</math>).</p> <ul style="list-style-type: none"> <li>▪ IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO.</li> <li>▪ OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.</li> </ul>                                                                                                                                                                                                                                                                                                   | 10'h0 | R_W |
| 18:0  | <p>Transfer Size (XferSize)</p> <p>This field contains the transfer size in bytes for the current endpoint. The transfer size (<math>XferSize</math>) = Sum of buffer sizes across all descriptors in the list for the endpoint.</p> <p>In Buffer DMA, the core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet.</p> <ul style="list-style-type: none"> <li>▪ IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO.</li> <li>▪ OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.</li> </ul> | 19'h0 | R_W |

**Note:** Note for Descriptor DMA

The maximum transfer size supported is 219 bytes

The maximum packet count supported =  $XferSize / \text{Max Packet Size}$

The Transfer Size can span across multiple descriptors

### 33.5.5.26 Device Endpoint-*n* DMA Address Register (DIEPDMA<sub>n</sub>/DOEPDMA<sub>n</sub>)

◇ Endpoint\_number:  $0 \leq n \leq 15$

◇ Offset for IN endpoints:  $914h + (\text{Endpoint\_number} * 20h)$

◇ Offset for OUT endpoints:  $B14h + (\text{Endpoint\_number} * 20h)$

These registers are implemented in RAM instead of flop-based implementation.

**Table 33-70 Device Endpoint-*n* DMA Address Register: DIEPDMA<sub>n</sub>/DOEPDMA<sub>n</sub>**

| Field | Description                                                                                                                                                                                                                                                                                                                        | Reset                                             | Acces |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|-------|
| 31:0  | <p>DMA Address (DMAAddr)</p> <p>Holds the start address of the external memory for storing or fetching endpoint data.</p> <p><b>Note:</b> For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP</p> | "X" if not programmed as the register is in SPRAM | R_W   |

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |  |  |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | <p>data packet in the memory is overwritten.</p> <p>This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.</p> <ul style="list-style-type: none"> <li>▪ When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field.</li> <li>▪ When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.</li> </ul> |  |  |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

### 33.5.5.27 Device Endpoint-*n* DMA Buffer Address Register (DIEPDMABn/DOEPDMABn)

- ✧ Endpoint\_number:  $0 \leq n \leq 15$
- ✧ Offset for IN endpoints:  $91Ch + (\text{Endpoint\_number} * 20h)$
- ✧ Offset for OUT endpoints:  $B1Ch + (\text{Endpoint\_number} * 20h)$

These fields are present only in case of Scatter/Gather DMA. These registers are implemented in RAM instead of flop-based implementation.

**Table 33-71 Device Endpoint-*n* DMA Buffer Address Register: DIEPDMABn/DOEPDMABn**

| Field | Description                                                                                                                                                                                                                                                                                  | Reset                                             | Access |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|--------|
| 31:0  | <p>DMA Buffer Address (DMABufferAddr)</p> <p>Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress.</p> <p>This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.</p> | "X" if not programmed as the register is in SPRAM | R_O    |

### 33.5.5.28 Device IN Endpoint Transmit FIFO Status Register (DTXFSTSn)

- ✧ Endpoint\_number:  $0 \leq n \leq 15$
- ✧ Offset for IN endpoints:  $918h + (\text{Endpoint\_number} * 20h)$

This read-only register contains the free space information for the Device IN endpoint Tx FIFO.

**Table 33-72 Device IN Endpoint Transmit FIFO Status Register: DTXFSTSn**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset        | Access |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
| 31:16 | Reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                    |              | RO     |
| 15:0  | <p>IN Endpoint Tx FIFO Space Avail (INEPTxFSpAvail)</p> <p>Indicates the amount of free space available in the Endpoint Tx FIFO. Values are in terms of 32-bit words.</p> <ul style="list-style-type: none"> <li>▪ 16'h0: Endpoint Tx FIFO is full</li> <li>▪ 16'h1: 1 word available</li> <li>▪ 16'h2: 2 words available</li> <li>▪ 16'h<math>n</math>: <math>n</math> words available (where <math>0 \leq n \leq 32,768</math>)</li> </ul> | Configurable | RO     |



---

|  |                                                                                                               |  |  |
|--|---------------------------------------------------------------------------------------------------------------|--|--|
|  | <ul style="list-style-type: none"><li>▪ 16'h8000: 32,768 words available</li><li>▪ Others: Reserved</li></ul> |  |  |
|--|---------------------------------------------------------------------------------------------------------------|--|--|

# Section 7

# BOOT

## 34 XBurst Boot ROM Specification

The JZ4780 contains an internal 16KB boot ROM. The CPU boots from the boot ROM after reset.

### 34.1 Boot Select

The boot sequence of the JZ4780 is controlled by boot\_sel [2:0]. The configuration is shown as follow:

**Table 34-1 Boot Configuration of JZ4780**

| boot_sel[2:0] | Boot method                              |
|---------------|------------------------------------------|
| 110           | NAND boot @ CS1                          |
| 101           | SD boot @ MSC0 ( MMC/SD use GPIO Port A) |
| 011           | eMMC boot @ MSC0 (use GPIO Port A)       |
| 000           | SPI boot @ SPI0/CE0                      |
| 010           | NOR boot @ CS4 (just for FPGA testing)   |
| 111           | USB boot @ USB 2.0 device, EXTCLK=12MHz  |
| 100           | SD boot @ MSC1 (MMC/SD use GPIO Port E)  |

### 34.2 Boot Procedure

After reset, the boot program on the internal boot ROM executes as follows:

- 1 Disable all interrupts and read boot\_sel[2:0] to determine the boot method.
- 2 If it is boot from NAND flash, 4 flags at the beginning of NAND are read to know the NAND information including nand type, page cycle(2 or 3 cycles) and its page size(512B, 2KB, 4KB 8KB or 16KB). Then 14KB code are read out from NAND to tcsm, if the 14KB reading failed, the next 14KB backup in NAND will be read. Then branch to tcsm at 192 bytes offset.
- 3 There 14KB backup reading failed, the 14KB backup at 128<sup>th</sup>, 256<sup>th</sup>, ..., and finally 1024<sup>th</sup> page will be tried in consecutive order.
- 4 If it is boot from MMC/SD card at MSC0, its function pins MSC0\_D0, MSC0\_CLK, MSC0\_CMD are initialized, the boot program loads the 14KB code from MMC/SD card to tcsm and jump to it. Only one data bus which is MSC0\_D0 is used. The clock EXTCLK/128 is used initially. When reading data, the clock EXTCLK/4 is used.
- 5 If it is boot from eMMC boot partition1 at MSC0, its function pins MSC0\_D0, MSC0\_CLK, MSC0\_CMD are initialized, the boot program loads the 14KB code from eMMC boot partition1 to tcsm and jump to it. Only one data bus which is MSC0\_D0 is used. The clock EXTCLK/4 is used.
- 6 If it is boot from USB, a block of code will be received through USB cable connected with host PC and be stored in tcsm. Then branch to this area in tcsm.
- 7 If it is boot from MMC/SD card at MSC1, its function pins MSC1\_D0,D1,D2,D3, MSC1\_CLK,

MSC1\_CMD are initialized, the boot program loads the 14KB code from MMC/SD card to tcsms and jump to it.

**NOTE:** The JZ4780's tcsms is 16KB, its address is from 0xf4000000 to 0xf4004000.

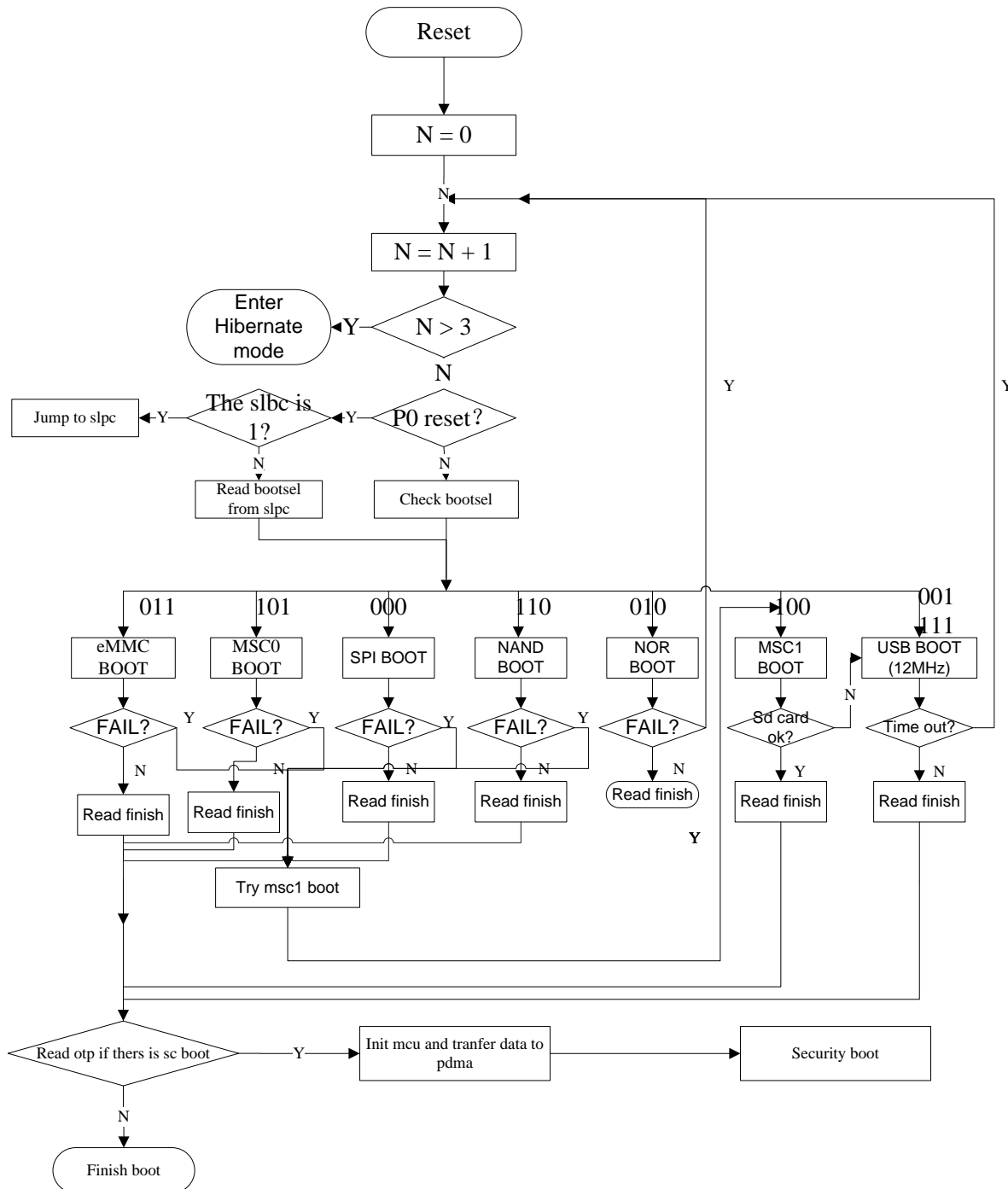


Figure 34-1 Boot sequence diagram of JZ4780

### 34.3 NAND Boot Specification

If CPU boots from NAND flash (CS1), the boot ROM will read 5 flags from NAND flash to know the NAND information including row cycle(2 or 3 ) and its page size(512, 2KB, 4KB, 8KB or 16KB bytes).

The content and definition of the 5 flags are shown as follow:

**Table 34-2 The definition of 5 flags in NAND flash**

| Name           | Location<br>(in byte) | length<br>(in byte) | Value              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                 |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
|----------------|-----------------------|---------------------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|-------|-----------------|------|------|------|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|
| nandtype_flag  | 0-63                  | 64                  | 0x55<br>or<br>0xaa | NAND type.<br>0x55: common nand<br>0xaa: toggle nand                                                                                                                                                                                                                                                                                                                                                                                                        |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
| rowcycle_flag  | 64-95                 | 32                  | 0x55<br>or<br>0xaa | The number of bytes of row cycles.<br>0x55: 2-byte row cycles<br>0xaa: 3-byte row cycles                                                                                                                                                                                                                                                                                                                                                                    |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
| pagesize_flag2 | 96-127                | 32                  | 0x55<br>or<br>0xaa | <table border="0"> <tr> <td>flag2</td> <td>flag1</td> <td>flag0</td> <td>pagesize(byte).</td> </tr> <tr> <td>0x55</td> <td>0x55</td> <td>0x55</td> <td>512</td> </tr> <tr> <td>0x55</td> <td>0xaa</td> <td>0x55</td> <td>2048</td> </tr> <tr> <td>0xaa</td> <td>0x55</td> <td>0x55</td> <td>4096</td> </tr> <tr> <td>0xaa</td> <td>0xaa</td> <td>0x55</td> <td>8192</td> </tr> <tr> <td>0xaa</td> <td>0xaa</td> <td>0xaa</td> <td>16384</td> </tr> </table> | flag2 | flag1 | flag0 | pagesize(byte). | 0x55 | 0x55 | 0x55 | 512 | 0x55 | 0xaa | 0x55 | 2048 | 0xaa | 0x55 | 0x55 | 4096 | 0xaa | 0xaa | 0x55 | 8192 | 0xaa | 0xaa | 0xaa | 16384 |
| flag2          | flag1                 | flag0               | pagesize(byte).    |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
| 0x55           | 0x55                  | 0x55                | 512                |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
| 0x55           | 0xaa                  | 0x55                | 2048               |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
| 0xaa           | 0x55                  | 0x55                | 4096               |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
| 0xaa           | 0xaa                  | 0x55                | 8192               |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
| 0xaa           | 0xaa                  | 0xaa                | 16384              |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
| pagesize_flag1 | 128-159               | 32                  | 0x55<br>or<br>0xaa |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |
| pagesize_flag0 | 160-191               | 32                  | 0x55<br>or<br>0xaa |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |       |       |       |                 |      |      |      |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |       |

The nandtype\_flag containing 64 bytes locates at the beginning of NAND, if the type of NAND is common, the nandtype\_flag should be filled with 0x55 for all 64 bytes, or else it should be filled with 0xaa. The rowcycle\_flag containing 32 bytes locates behind the nandtype\_flag, if the number of bytes of row cycles is 2, the flag should be filled with 0x55 for all 32 bytes, or else it should be filled with 0xaa. The pagesize\_flag2 , pagesize\_flag1 and pagesize\_flag0 each containing 32 bytes locate behind the rowcycle\_flag, which value should be filled is determined by the page size of NAND. Please refer to above table. Totally, 192 bytes are allocated for the 5 flags.

At first, the first 256 bytes (which is a PN\* unit) in NAND containing 5 flags will be read out to a buffer assuming the nandtype is common. The nandtype\_flag will be get from the buffer to detect the nand type. If there is no 0x55 or 0xaa in nandtype\_flag, 128<sup>th</sup>, 256<sup>th</sup>, ..., 1024<sup>th</sup> page will be tried in sequence. If failed at 1024<sup>th</sup> page, bootrom will jump to sd\_boot. If nandtype\_flag is valid, the rowcycle\_flag will be obtained from the buffer to know the number of row cycles. At last, pagesize\_flag2 ,pagesize\_flag1 and pagesize\_flag0 will be obtained from the buffer to know precise page size.

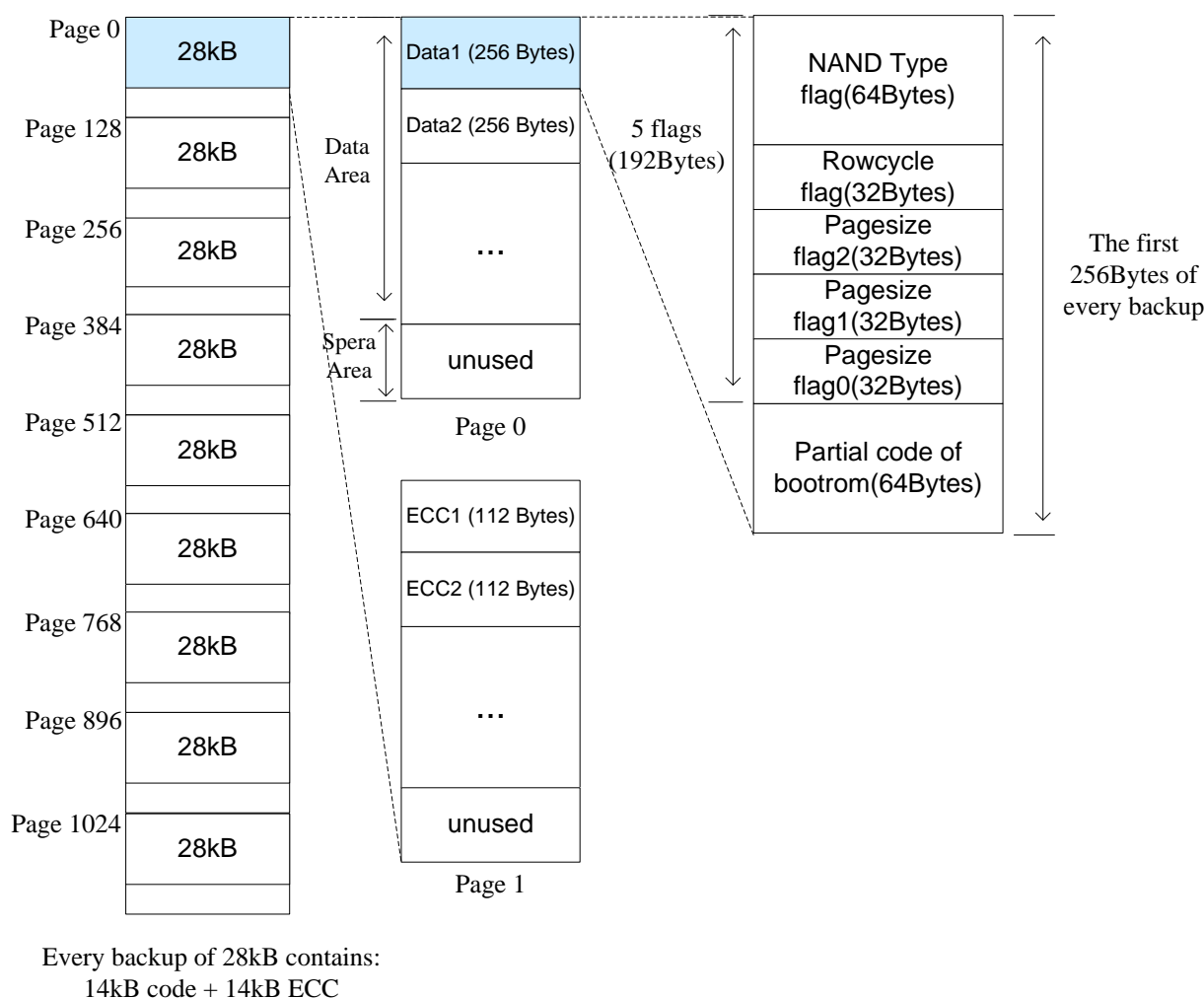
14KB codes in NAND will be loaded up to tcsn and branch to it at 192 bytes offset. Hardware PN and 64-bit BCH ECC will be used for every 256 bytes during reading. The ECC(112 bytes per 256 bytes

data) stores in the data area of a NAND page behind the page storing code data. If no ECC error is detected or ECC error is correctable(number of error bits <= 64), NAND boots successfully. If uncorrectable error occurred, next 14KB backup at 128 pages behind will be tried. 128<sup>th</sup>, 256<sup>th</sup>, ..., 1024<sup>th</sup> page will be tried in sequence. If failed at 1024<sup>th</sup> page, bootrom will jump to sd boot(msc1).

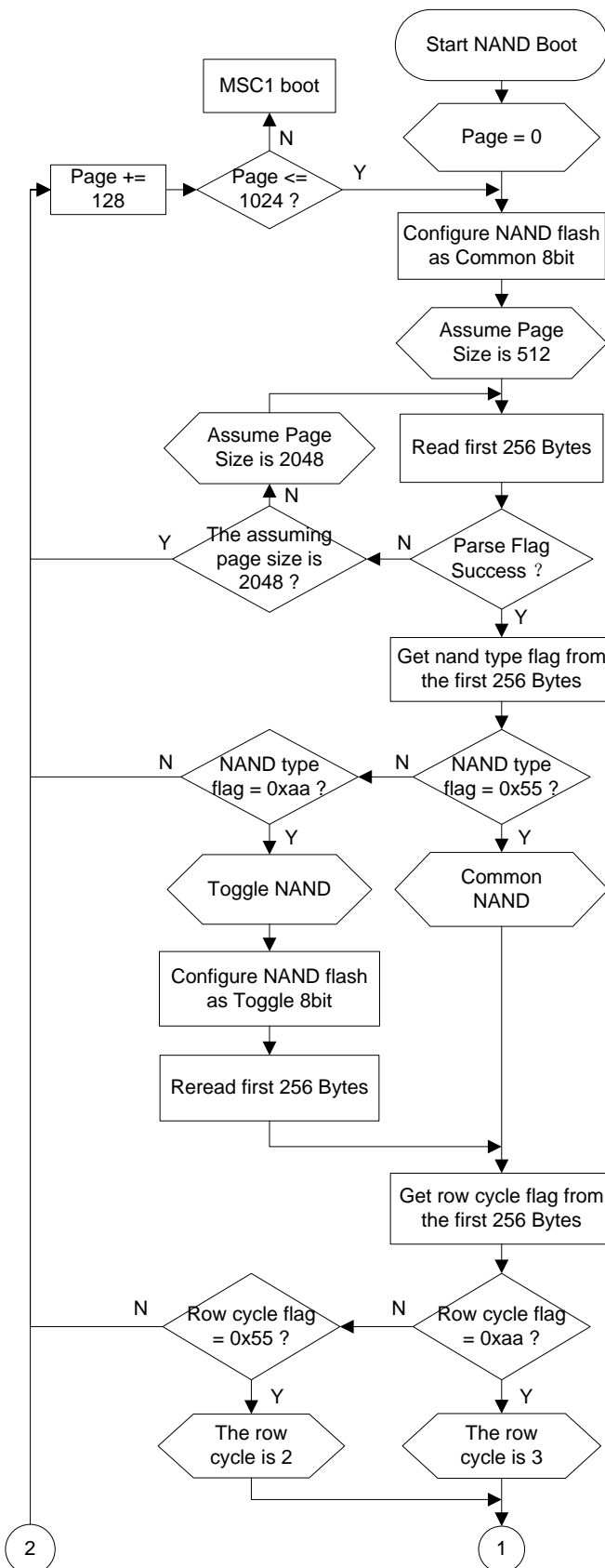
The distribution and structure of the boot code in NAND is shown as Figure 36-2.

The procedure of the JZ4780 NAND boot is shown as Figure 34-3.

**NOTE:** PN is short for pseudorandom noise which is used for supporting TLC ( three-level cell ) NAND.



**Figure 34-2 the distribution and structure of the boot code in NAND**



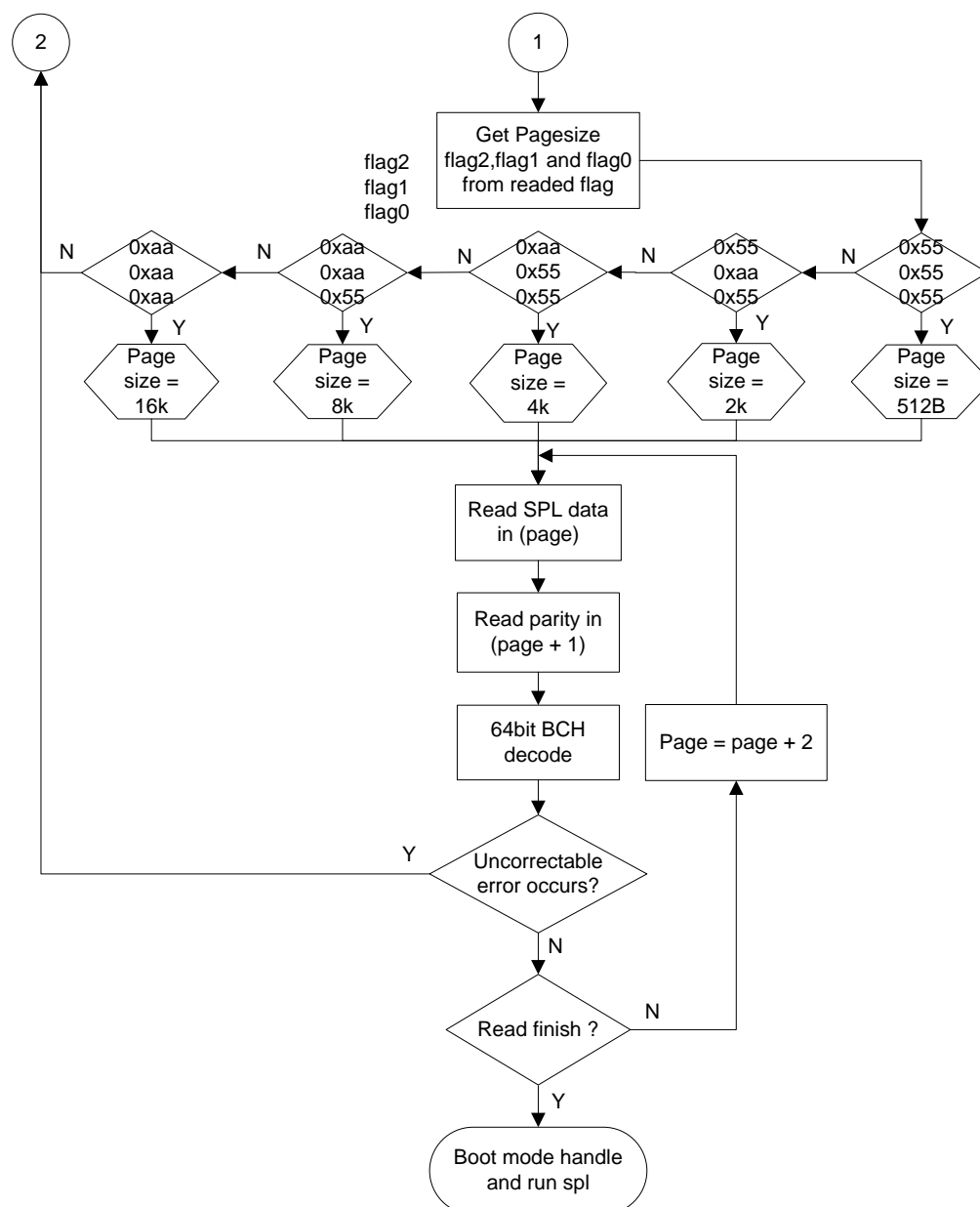


Figure 34-3 JZ4780 NAND Boot Procedure

### 34.4 USB Boot Specification

When boot\_sel[2:0] is selected as USB boot, the internal boot ROM downloads user program from the USB port to internal SRAM and branches to the internal SRAM to execute the program.

JZ4780 supports the external main crystal whose frequency is 12MHz.

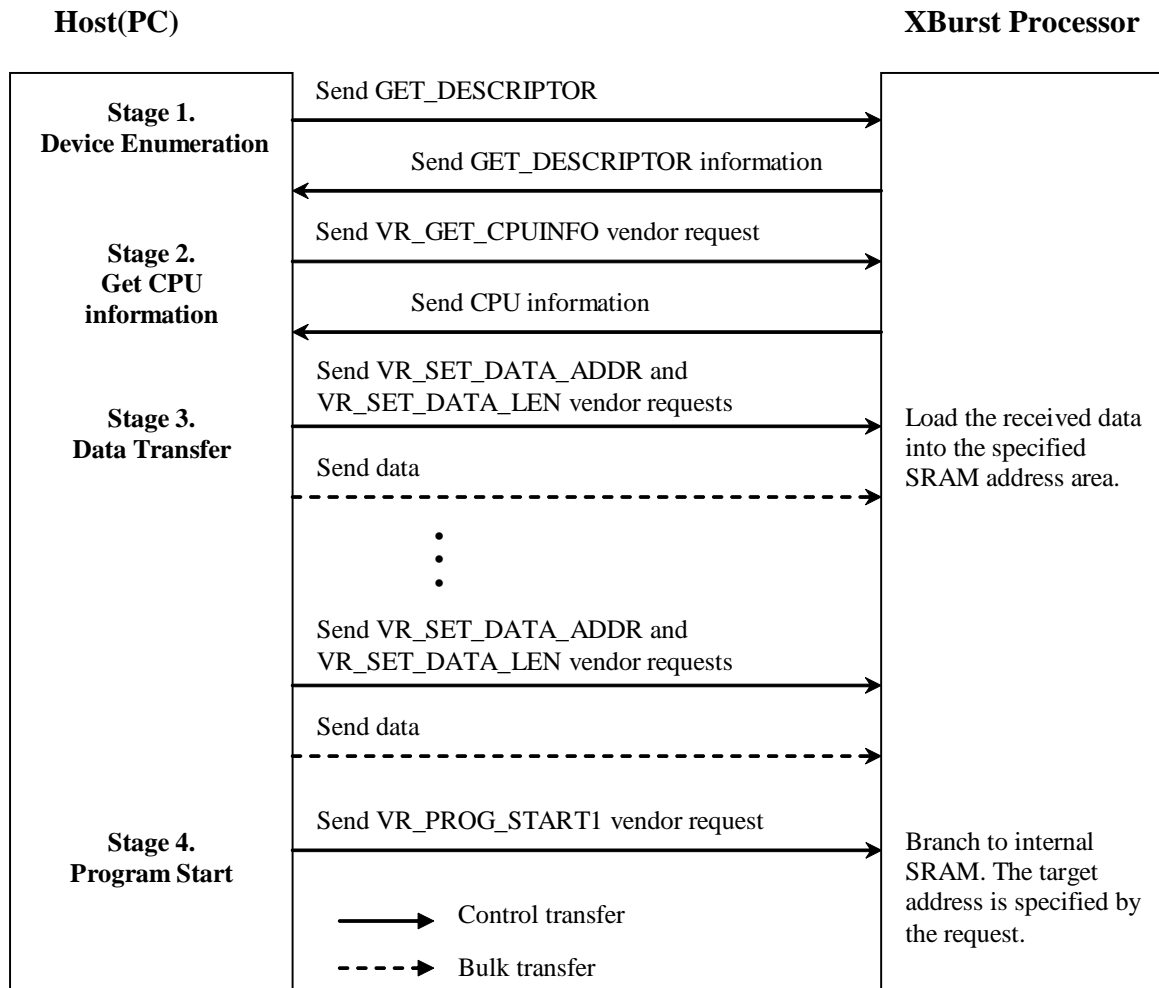
The boot program supports both high-speed (480MHz) and full-speed (12MHz) transfer modes. The boot program uses the following two transfer types.



**Table 34-3 Transfer Types Used by the Boot Program**

| Transfer Type    | Description                                                             |
|------------------|-------------------------------------------------------------------------|
| Control Transfer | Used for transmitting standard requests and vendor requests.            |
| Bulk Transfer    | Used for responding to vendor requests and transmitting a user program. |

The following figure shows an overview of the USB communication flow.



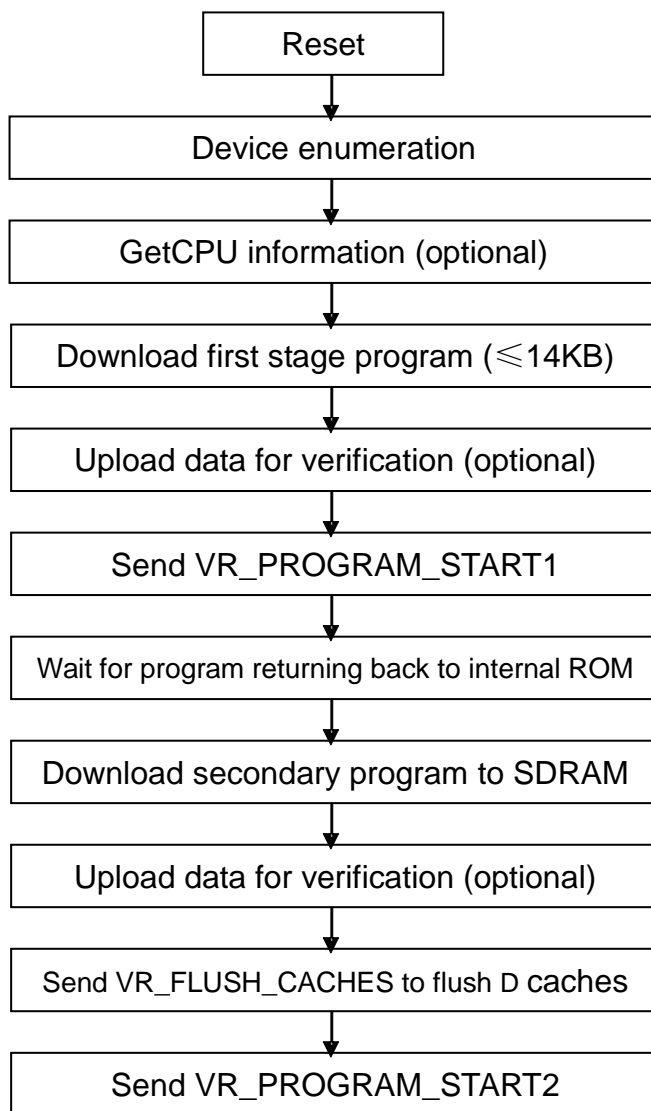
**Figure 34-4 USB Communication Flow**

The vendor ID and product ID for the USB boot device are 0xa108 and 0x4780 respectively. The Configuration for USB is for Control Endpoint 0 with Max Packet Size equals 64 bytes, Bulk IN at Endpoint 1 with Max Packet Size equals 512 bytes in high-speed and 64 bytes in full-speed, Bulk OUT at Endpoint 1 with Max Packet Size equals 512 bytes in high-speed and 64 bytes in full-speed.

The USB boot program provides six vendor requests through control endpoint for user to download/upload data to/from device, and to branch to a target address to execute user program. The six vendor requests are `VR_GET_CPU_INFO` (0x00), `VR_SET_DATA_ADDRESS` (0x01),

VR\_SET\_DATA\_LENGTH (0x02), VR\_FLUSH\_CACHES (0x03), VR\_PROGRAM\_START1 (0x04) and VR\_PROGRAM\_START2 (0x05). User program is transferred through Bulk IN or Bulk OUT endpoint.

When JZ4780 is reset with boot\_sel[2:0] equals 110b, 001b or 010b, the internal boot ROM will switch to USB boot mode and wait for USB requests from host. After connecting the USB device port to host, host will recognize the connection of a USB device, and start device enumeration. After finishing the device enumeration, user can send VR\_GET\_CPU\_INFO (0x00) to query the device CPU information. If user wants to download/upload a program to/from device, two vendor requests VR\_SET\_DATA\_ADDRESS (0x01) and VR\_SET\_DATA\_LENGTH (0x02) should be sent first to tell the device the address and length in byte of the subsequent transferring data. Then data can be transferred through bulk-out/bulk-in endpoint. After this first stage program has been transferred to device, user can send vendor request VR\_PROGRAM\_START1 (0x04) to let the CPU to execute the program. This first stage program must not greater than 14KB and is normally used to init GPIO and SDRAM of the target board. At the end of the first stage program, it can return back to the internal boot ROM by jumping to ra (\$31) register. Thus user can download a new program to the SDRAM of the target board like the first stage, and send vendor request VR\_FLUSH\_CACHES (0x03) and VR\_PROGRAM\_START2 (0x05) to let the CPU to execute the new program. Next figure is the typical procedure of USB boot.



**Figure 34-5 Typical Procedure of USB Boot**

Following tables list all the vendor requests that USB boot program supports:

**Table 34-4 Vendor Request 0 Setup Command Data Structure**

| Offset | Field         | Size | Value | Description                                                   |
|--------|---------------|------|-------|---------------------------------------------------------------|
| 0      | bmRequestType | 1    | 40H   | D7 0: Host to Device.<br>D6-D5 2: Vendor.<br>D4-D0 0: Device. |
| 1      | bRequest      | 1    | 00H   | VR_GET_CPU_INFO: get CPU information.                         |
| 2      | wValue        | 2    | 0000H | Not in used.                                                  |
| 4      | wIndex        | 2    | 0000H | Not in used.                                                  |
| 6      | wLength       | 2    | 0008H | 8 bytes.                                                      |

**Table 34-5 Vendor Request 1 Setup Command Data Structure**

| Offset | Field         | Size | Value | Description                                                          |
|--------|---------------|------|-------|----------------------------------------------------------------------|
| 0      | bmRequestType | 1    | 40H   | D7 0: Host to Device.<br>D6-D5 2: Vendor.<br>D4-D0 0: Device.        |
| 1      | bRequest      | 1    | 01H   | VR_SET_DATA_ADDRESS: set address for next bulk-in/bulk-out transfer. |
| 2      | wValue        | 2    | xxxxH | MSB (bit[31:16]) of the data address.                                |
| 4      | wIndex        | 2    | xxxxH | LSB (bit[15:0]) of the data address.                                 |
| 6      | wLength       | 2    | 0000H | Not in used.                                                         |

**Table 34-6 Vendor Request 2 Setup Command Data Structure**

| Offset | Field         | Size | Value | Description                                                                |
|--------|---------------|------|-------|----------------------------------------------------------------------------|
| 0      | bmRequestType | 1    | 40H   | D7 0: Host to Device.<br>D6-D5 2: Vendor.<br>D4-D0 0: Device.              |
| 1      | bRequest      | 1    | 02H   | VR_SET_DATA_LENGTH: set length in byte for next bulk-in/bulk-out transfer. |
| 2      | wValue        | 2    | xxxxH | MSB (bit[31:16]) of the data length.                                       |
| 4      | wIndex        | 2    | xxxxH | LSB (bit[15:0]) of the data length.                                        |
| 6      | wLength       | 2    | 0000H | Not in used.                                                               |

**Table 34-7 Vendor Request 3 Setup Command Data Structure**

| Offset | Field         | Size | Value | Description                                                   |
|--------|---------------|------|-------|---------------------------------------------------------------|
| 0      | bmRequestType | 1    | 40H   | D7 0: Host to Device.<br>D6-D5 2: Vendor.<br>D4-D0 0: Device. |
| 1      | bRequest      | 1    | 03H   | VR_FLUSH_CACHES: flush I-Cache and D-Cache.                   |
| 2      | wValue        | 2    | 0000H | Not in used.                                                  |
| 4      | wIndex        | 2    | 0000H | Not in used.                                                  |
| 6      | wLength       | 2    | 0000H | Not in used.                                                  |

**Table 34-8 Vendor Request 4 Setup Command Data Structure**

| Offset | Field         | Size | Value | Description                                                   |
|--------|---------------|------|-------|---------------------------------------------------------------|
| 0      | bmRequestType | 1    | 40H   | D7 0: Host to Device.<br>D6-D5 2: Vendor.<br>D4-D0 0: Device. |

|   |          |   |       |                                                                                                                                                                                                                                                                                                                                                                                                    |
|---|----------|---|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | bRequest | 1 | 04H   | VR_PROGRAM_START1: transfer data from D-Cache to I-Cache and branch to address in I-Cache.<br><b>NOTE:</b> After downloading program from host to device for the first time, you can only use this request to start the program. Since the USB boot program will download data to D-Cache after reset. This request will transfer data from D-Cache to I-Cache and execute the program in I-Cache. |
| 2 | wValue   | 2 | xxxxH | MSB (bit[31:16]) of the program entry point.                                                                                                                                                                                                                                                                                                                                                       |
| 4 | wIndex   | 2 | xxxxH | LSB (bit[15:0]) of the program entry point.                                                                                                                                                                                                                                                                                                                                                        |
| 6 | wLength  | 2 | 0000H | Not in used.                                                                                                                                                                                                                                                                                                                                                                                       |

**Table 34-9 Vendor Request 5 Setup Command Data Structure**

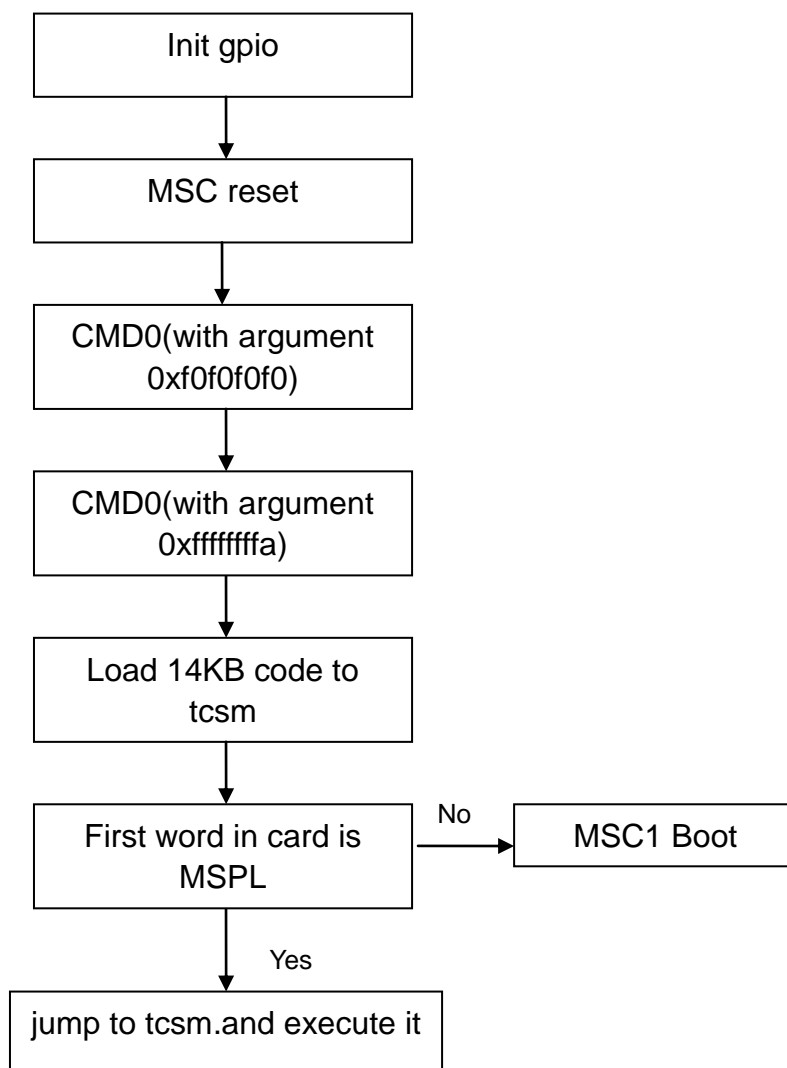
| Offset | Field         | Size | Value | Description                                                   |
|--------|---------------|------|-------|---------------------------------------------------------------|
| 0      | bmRequestType | 1    | 40H   | D7 0: Host to Device.<br>D6-D5 2: Vendor.<br>D4-D0 0: Device. |
| 1      | bRequest      | 1    | 05H   | VR_PROGRAM_START2: branch to target address directly.         |
| 2      | wValue        | 2    | xxxxH | MSB (bit[31:16]) of the program entry point.                  |
| 4      | WIndex        | 2    | xxxxH | LSB (bit[15:0]) of the program entry point.                   |
| 6      | WLength       | 2    | 0000H | Not in used.                                                  |

### 34.5 MSC0 Boot Specification

All cards can boot from MMC/SD Boot from MSC0, the boot program will load 14KB code starting at sector 0 from MMC/SD card to tcsm. First the boot program initializes MSC0\_D0, MSC0\_CLK, MSC0\_CMD as function pins. Only one data pin MSC0\_D0 is used. Then the boot program sends CMD55 to test if it's SD or MMC card and initializes the card. At last it loads 14KB code from the card to tcsm and branches to execute the code in tcsm.

When initializing the card, the clock of EXTCLK/128 is used. And when reading data, the clock of EXTCLK/4 is used.

The procedure of the JZ4780 MMC/SD boot is shown as follow:



### 34.6 eMMC Boot Specification

If eMMC is MultiMediaCard System Specification Ver. 4.4 compatible, you can use eMMC boot method. you should write boot code to boot partition1 , then the boot program will load 14KB code from eMMC boot partition1 area to tcsm. First the boot program initializes MSC0\_D0, MSC0\_CLK, MSC0\_CMD as function pins. Only one data pin MSC0\_D0 is used. Then the boot program sends CMD0 to set eMMC in boot mode. At last it loads 14KB code from the card to tcsm and branches to execute the code in tcsm. and the clock of EXTCLK/4 is used.

The procedure of the JZ4780 eMMC boot is shown as follow:

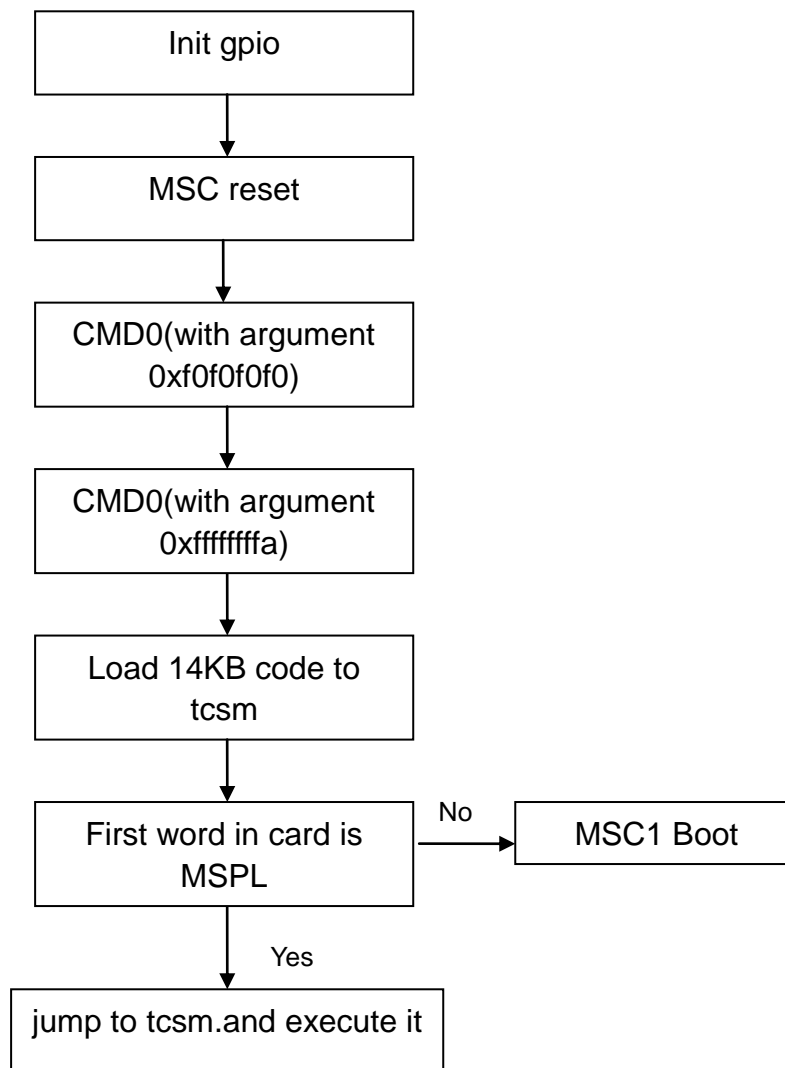


Figure 34-6 JZ4780 eMMC Boot Procedure

### 34.7 msc1 boot Specification

It is the same as msc0 boot. The purpose of msc1 boot is that,you can use it to burn other devices.

### 34.8 Spi boot Specification

When boot\_sel[2:0] is selected as SPI boot(boot\_sel = 000), CPU will boot from SPI nor flash by General-Purpose I/O Port-A 18, 20, 21, 23 pin. In spi nor flash address 0x0~0xF, this space will store 16 bytes that agreed SPI\_boot flag. SPI\_boot flag shown “SPI nor flash boot flag informations” table.

SPI\_boot detailed process shown “SPI Boot Procedure”

**Note:** Any irregularity in the above steps, SPI\_boot will disable SSI controller and jump to MSC1 boot.

The SPI\_boot flag information table and procedure of the JZ4780 SPI boot is shown as follow:

**Table 34-10 SPI nor flash boot flag informations**

|        |           |      |      |           |             |       |       |             |
|--------|-----------|------|------|-----------|-------------|-------|-------|-------------|
| addr   | 0x00      | 0x01 | 0x02 | 0x03      | 0x04        | 0x05  | 0x06  | 0x07        |
| detail | 0x04      | 0x03 | 0x02 | 0x55      | 0xaa        | 0x55  | 0xaa  | SSI_GR      |
| addr   | 0x08      | 0x09 | 0x0A | 0x0B      | 0x0C        | 0x0D  | 0x0E  | 0x0F        |
| detail | Len (LSB) | Len  | Len  | Len (MSB) | Check (LSB) | Check | Check | Check (MSB) |

**Explain:**

- 0x00~0x06 :Must store 0x04,0x03,0x02,0x55,0xAA,0x55,0xAA..
- 0x07 :The value of SSI Clock Generator Register.
- 0x08~0x0B :The length of copy data.
- 0x0C~0x0F :Copy data checksum.



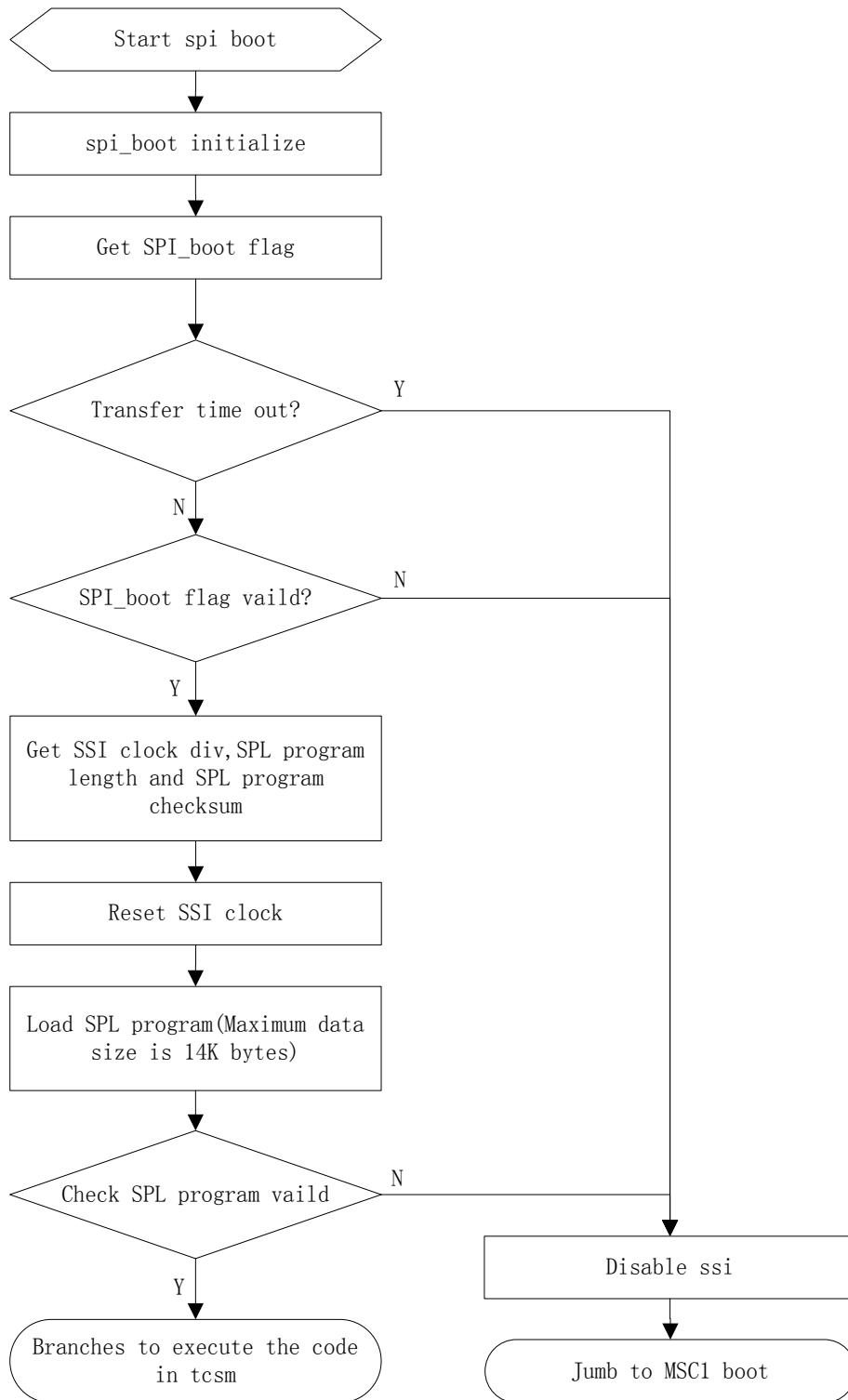


Figure 34-7 JZ4780 SPI Boot Procedure