# POWER8 Processor SCM and Memory Buffer Hardware Errata Notice

## DD 2.x

## Version 1.0

**Advance**

16 May 2014



Printed in the United States of America May 2014



IBM Systems and Technology Group
2070 Route 52, Bldg. 330
Hopewell Junction, NY 12533-6351

The IBM home page can be found at **ibm.com®**.

Version 1.0
16 May 2014

## Introduction

For the design revision levels specified (DD 2.x), this document identifies differences between the actual operation of the POWER8 Processor SCM and POWER8 Memory Buffer devices and what is described in the POWER8 User Manuals and Datasheets.

Differences that are not visible to the user are not described. If a difference is hidden or compensated for by IBM-supplied firmware or by IBM-supplied system software, it is not included in this document.

This is a working document. Check regularly with your IBM technical representative to verify that you have the most current version. For more information, contact OpenPOWER@us.ibm.com.

**Note:** Some of the workarounds described in this document might involve the use of IBM software or firmware modules to affect or reduce the effect of the erratum. These modules are typically supplied with the POWER8 devices. If, however, you do not have the module required for a workaround, contact IBM to acquire it.

### Revision Levels Covered

This document includes information about errata that apply to the following design revision levels:

- POWER8 Processor SCM revision DD 2.0 and later
- POWER8 Memory Buffer revision DD 2.0 and later

This document does not contain information about POWER8 Processor or POWER8 Memory Buffer levels before DD 2.0. Any statements in this document about other revision levels of these products are for reference only. For errata information about these revision levels, see the current Errata Notice for the specific revision of the product.

### Errata Categories

Each erratum is assigned to a category numbered 1 - 5 based on its impact on system performance and the ability of any proposed workaround to minimize that impact. *Table 1* explains the meaning of each category.

*Table 1. Errata Categories*

| Errata Category | Definition |
|---|---|
| 1 | Major impact; no workaround available. An erratum is said to have a major impact if it results in a system crash, a hard failure, an unrecoverable soft failure, significant performance degradation, or the storage of incorrect data. |
| 2 | Major impact; workaround is impractical to implement, or a substantial risk exists of encountering the same problem or additional problems, including performance issues, after the workaround is implemented. |
| 3 | Major impact; workaround available. Application of the workaround either eliminates the problem, or reduces it to a minor impact issue. |
| 4 | Minor impact; no workaround available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. |
| 5 | Minor impact; workaround is available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. Application of the workaround eliminates the problem. |

## Terminology

The following section headings and terms are used in this document:

| | |
|---|---|
| Design Notes | Design notes are clarifications or additions to the published documentation about topics that IBM considers of special interest. |
| Erratum #n | The errata are numbered sequentially. |
| Erratum | An erratum is any difference between the actual operation of the specific design revision level and the design specification. |
| Abstract | The abstract is a brief description of the erratum. |
| Impact | A category is assigned to an erratum based on its impact on system performance and the ability of any proposed workaround to minimize that impact. See *Table 1* on page 3. |
| Published Date | The date that the erratum was first published. |
| Applies to | Indicates which devices and revision levels are affected by the erratum. |
| Status | This is the current status of the erratum discussion and the current plan for the future. |
| Description | This section provides a description of the erratum. |
| Workarounds | This section lists any actions that customers can take to reduce or eliminate the effects of the erratum. These actions supplement any workarounds embedded in the IBM code. If you are not using the operating system or firmware supplied or recommended by IBM, contact OpenPOWER@us.ibm.com. |

## Summary of Errata

*Table 2* summarizes the errata described in this document.

*Table 2. Summary of Errata*

| Erratum Number | Abstract | Errata Category | Workaround Available? | See Page |
|---|---|---|---|---|
| 1 | HID4 bit 33 must remain static after boot. | 5 | Yes | 6 |
| 2 | NX CRB reported as both killed and completed. | 5 | Yes | 7 |
| 3 | Incorrect data during scrub and steer cleanup with UE trap mode enabled. | 5 | Yes | 8 |
| 4 | Cfam_reset after $V_{IO}$ power on does not trigger selfboot. | 5 | Yes | 9 |
| 5 | Multithreaded XSCOM causes false Done and delayed error handling. | 5 | Yes | 10 |
| 6 | Certain software sequences result in ignored PMU interrupts. | 5 | Yes | 11 |
| 7 | Micropartition prefetch incompatible with 2 LPAR and 4 LPAR mode. | 5 | Yes | 13 |
| 8 | During a PMU state save routine, MMCR0[PMAO] and MMCR0[PMAE] are inconsistent for one cycle on every counter overflow while exceptions are enabled. | 5 | Yes | 14 |

## Design Note 1

**Abstract:** Powering off $V_{CS}$ during IPL is not supported.

**Published Date:** 16 May 2014

**Applies To:** POWER8 Memory Buffer

**Status:** No fix is planned.

### Description

System power dissipation can be reduced by disabling the L4 cache and turning off $V_{CS}$ when not in use. However, $V_{CS}$ must be on during system power on or IML/IPL (specifically during scan ring rotation), because some of the logic that is powered by $V_{CS}$ is required for this function.

### Workarounds

None.

## Erratum #1

**Abstract:** HID4 bit 33 must remain static after boot.

**Impact:** Category 5.
Minor impact; workaround is available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. Application of the workaround eliminates the problem.

**Published Date:** 16 May 2014

**Applies To:** POWER8 Processor SCM

**Status:** No fix is planned.

### Description

Changing the state of HID4[33] RMSC_EN with an **mtspr** IBM PowerPC® instruction can cause a core checkstop.

### Workarounds

Set the state of HID4[33] RMSC_EN at POR scan time and do not change it.

## Erratum #2

**Abstract:** NX CRB reported as both killed and completed.

**Impact:** Category 5.
Minor impact; workaround is available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. Application of the workaround eliminates the problem.

**Published Date:** 16 May 2014

**Applies To:** POWER8 Processor SCM

**Status:** No fix is planned.

### Description

This erratum is triggered when all of the following conditions are met:

1. Two cop_req[1] commands for the same NX coprocessor type (CT) are received in the same cycle on NX ports rcmd0 and rcmd1.
2. The destination CT engines and the DMA channels servicing those engines are busy with older jobs so the NX Powerbus interface (PBCQ) uses at least one floating queue slot to store one of the received cop_req CRBs.
3. A CRB Kill ISN or LPID is issued and hits the more recent job in a floating queue.

This erratum can cause the job in the floating queue to be reported as both killed and also completed. A job should be reported as killed or completed, but not both. The erratum can also cause an arbitrary job to be executed twice, either on the same or different CT.

### Workarounds

The workaround is to align CRBs on 256-byte boundaries. This forces all cop_req to the NX rcmd0 port. Therefore, trigger condition 1 cannot occur. The PHYP already aligns CRBs on 4 KB boundaries, so there is no impact.

1. A cop_req is the Powerbus command that results from an **icswx.** instruction. It has a 128-byte operand called a CRB that is pushed to the NX some time after the cop_req command is issued to the bus.

## Erratum #3

**Abstract:** Incorrect data during scrub and steer cleanup with UE trap mode enabled.

**Impact:** Category 5.
Minor impact; workaround is available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. Application of the workaround eliminates the problem.

**Published Date:** 16 May 2014

**Applies To:** POWER8 Memory Buffer

**Status:** No fix is planned.

### Description

When performing a scrub or steer cleanup maintenance command while UE trap mode is enabled, the write back data is incorrect.

**Note:** IBM hardware procedures only enable UE trap mode during the Superfast Read command that is used during IPL memory diagnostics. This traps the actual UE data so that it can be compared to the known pattern that was written to identify which bits contributed to the UE. UE trap mode is not used during normal run-time operation.

### Workarounds

Do not enable UE trap mode when performing scrub or steer cleanup operations.

## Erratum #4

**Abstract:** Cfam_reset after $V_{IO}$ power on does not trigger selfboot.

**Impact:** Category 5.
Minor impact; workaround is available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. Application of the workaround eliminates the problem.

**Published Date:** 16 May 2014

**Applies To:** POWER8 Processor SCM

**Status:** No fix is planned.

### Description

In a selfboot environment, the SCAN0 in ClockControl is not triggered after cfam_reset_b.

### Workarounds

After a cfam_reset, a manual trigger is required. This is done by providing the appropriate pattern at the flexible service interface (FSI).

## Erratum #5

**Abstract:** Multithreaded XSCOM causes a false Done and delayed error handling.

**Impact:** Category 5.
Minor impact; workaround is available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. Application of the workaround eliminates the problem.

**Published Date:** 16 May 2014

**Applies To:** POWER8 Processor SCM

**Status:** No fix is planned.

### Description

During XSCOM operations, both of the following events can occur:

- The XSCOM error status on thread0 is delayed by one cycle relative to Done.
- Multithreaded XSCOMs can cause a false Done indication on thread0 HMER.

### Workarounds

The workaround involves two steps:

1. Read the HMER a second time after Done to get the correct status.
2. Ensure that there is a single-threaded XSCOM on a per-core level via a software mutex.

## Erratum #6

**Abstract:** Certain software sequences result in ignored PMU interrupts.

**Impact:** Category 5.
Minor impact; workaround is available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. Application of the workaround eliminates the problem.

**Published Date:** 16 May 2014

**Applies To:** POWER8 Processor SCM

**Status:** No fix is planned.

### Description

PMU interrupts are raised to the core through an internal latch that is asserted exclusively when either of the following events occurs:

- PMC1 overflows while MMCR0[PMAE] and MMCR0[PMC1CE] are enabled.
- At least one of PMC2 - PMC6 overflows while MMCR0[PMAE] and MMCR0[PMCjCE] are enabled.

The following four sequences fail to produce a PMU interrupt:

1. PMC1 is negative, followed by both MMCR0[PMAE] and MMCR0[PMC1CE] being enabled, in any order.
2. At least one of PMC2 - PMC6 is negative, followed by both MMCR0[PMAE] and MMCR0[PMCjCE] being enabled, in any order.
3. Both MMCR0[PMAE] and MMCR0[PMC1CE] are enabled, followed by software writing a negative value to PMC1.
4. Both MMCR0[PMAE] and MMCR0[PMCjCE] are enabled, followed by software writing a negative value to at least one of PMC2 - PMC6.

### Workarounds

All three of the following actions must be taken to work around the erratum:

1. Before enabling random sampling (by asserting MMCR0[PMAE] and at least one of MMCR0[PMC1CE] or MMCR0[PMCjCE]), clear the PMCs.
2. Do not write negative values to any of the PMCs outside of save/restore routines.
3. Before performing a PMU state restore, check MMCR0 and the PMC values captured by the save routine.

   **Note:** This step assumes that the previous state of the PMU was saved as copies of the PMC*n*, MMCR0, SIER, SIAR, SDAR, and MMCR2 registers that will be restored.

   If either of the following conditions is true, follow the *Workaround Sequence* on page 12 while the External Interrupt Enable (MSR[EE]) bit is 0. Otherwise, follow the normal restore routine.

   - PMC1 is negative, MMCR0[PMAO] is clear, and both MMCR0[PMAE] and MMCR0[PMC1CE] are enabled.

- At least one of PMC2 - PMC6 is negative, MMCR0[PMAO] is clear, and both MMCR0[PMAE] and MMCR0[PMCjCE] are enabled.

*Workaround Sequence*

1. Write 0x0000000006004100 to MMCR0 (the actual register, not the saved copy).
2. Write 0x0000000000000000 to MMCR2 (the actual register, not the saved copy).
3. Write 0x7FFFFFFF to PMC6 (the actual register, not the saved copy).
4. Restore the saved copies of the PMCs.
5. Restore the saved copies of SIER, SIAR, and SDAR.
6. Restore the saved copy of MMCR2.
7. Restore the saved copy of MMCR0.
8. Continue.

This sequence checks for a counter negative condition on the running thread. If there is one, instead of restoring the negative PMC with with MMCR0[PMAO] set to zero and ignoring the interrupt, it forces a new interrupt on PMC6. Thus, when this hypervisor sequence returns and the EE bit is restored to 1, the hardware takes the new interrupt using the restored data as if it was the original counter negative condition before random sampling was enabled.

## Erratum #7

**Abstract:** Micropartition prefetch incompatible with 2 LPAR and 4 LPAR mode.

**Impact:** Category 5.
Minor impact; workaround is available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. Application of the workaround eliminates the problem.

**Published Date:** 16 May 2014

**Applies To:** POWER8 Processor SCM

**Status:** No fix is planned.

### Description

The micropartition prefetch mechanism does not perform the desired partition data prefetch in 2 LPAR or 4 LPAR mode.

In 2 LPAR or 4 LPAR mode, incomplete and or excessive prefetching can occur. If a micropartition prefetch is performed in 2 LPAR or 4 LPAR mode, prefetch state machines are left in a state that makes them unavailable for subsequent data prefetch operations.

The micropartition pefetch mechanism functions correctly only in 1 LPAR mode.

### Workarounds

System partition management software must not activate micropartition prefetch through the MPPR register if the POWER8 processor is running in 2 LPAR or 4 LPAR mode.

## Erratum #8

**Abstract:** During a PMU state save routine, MMCR0[PMAO] and MMCR0[PMAE] are inconsistent for one cycle on every counter overflow while exceptions are enabled.

**Impact:** Category 5. Minor impact; workaround is available. An erratum is said to have a minor impact if it results in slight to moderate performance degradation or is a functional variance from the specification. Application of the workaround eliminates the problem.

**Published Date:** 16 May 2014

**Applies To:** POWER8 Processor SCM

**Status:** No fix is planned

### Description

PMU interrupts are raised to the core through an internal latch that is asserted exclusively when either of the following events occurs:

- PMC1 overflows while MMCR0[PMAE] and MMCR0[PMC1CE] are enabled.
- At least one of PMC2 - PMC6 overflows while MMCR0[PMAE] and MMCR0[PMCjCE] are enabled.

It takes one cycle for MMCR0[PMAO] to be set and MMCR0[PMAE] to be cleared after the overflow happens. It is possible for software to read MMCR0 in this inconsistent state at the time of a PMU state save routine and potentially drop the pending interrupt.

### Workaround

Before performing a PMU state restore, check MMCR0 and the PMC values captured by the save routine.

**Note:** This workaround assumes that the previous state of the PMU was saved as copies of the PMC*n*, MMCR0, SIER, SIAR, SDAR, and MMCR2 registers that will be restored.

If either of the following conditions is true, follow the *Workaround Sequence* on page 14 while the External Interrupt Enable (MSR[EE]) bit is 0. Otherwise, follow the normal restore routine.

- PMC1 is negative, MMCR0[PMAO] is clear, and both MMCR0[PMAE] and MMCR0[PMC1CE] are enabled.
- At least one of PMC2 - PMC6 is negative, MMCR0[PMAO] is clear, and both MMCR0[PMAE] and MMCR0[PMCjCE] are enabled.

*Workaround Sequence*

1. Write 0x0000000006004100 to MMCR0 (the actual register, not the saved copy).
2. Write 0x0000000000000000 to MMCR2 (the actual register, not the saved copy).
3. Write 0x7FFFFFFF to PMC6 (the actual register, not the saved copy).
4. Restore the saved copies of the PMCs.
5. Restore the saved copies of SIER, SIAR, and SDAR.

6. Restore the saved copy of MMCR2.
7. Restore the saved copy of MMCR0.
8. Continue.

This sequence checks for a counter negative condition on the running thread. If there is one, instead of restoring the negative PMC with with MMCR0[PMAO] set to zero and ignoring the interrupt, it forces a new interrupt on PMC6. Thus, when this hypervisor sequence returns and the EE bit is restored to 1, the hardware takes the new interrupt using the restored data as if it was the original counter negative condition.

# Revision Log

Each release of this document supersedes all previously released versions. The revision log lists all significant changes made to the document since its initial release. In the rest of the document, change bars in the margin indicate that the adjacent text was modified from the previous release of this document.

| Revision Date | Description |
|---|---|
| 16 May 2014 | Version 1.0 (initial release). |