Introduction

This document contains errata that affect designs using the Ampere Computing® AmpereOne® devices.

Each erratum includes an overview, a description of the system impact, and a description of possible workaround(s).

Refer to *Table 2, "Errata Summary", on page 3* for the list of errata.

Errata are organized by the item designator in an alphabetical order. The item designator consists of an acronym for the affected functional unit and a numeric value. Numeric values are assigned to all errata.

List of Functional Unit Acronyms

- CPU — AmpereOne Processing Element (PE) within the Processor Complex (PCP)
- DEBUG — Debug Access Port (DAP) and other Debug Components
- MCU — Memory Controller Unit
- MESH — Coherent Mesh Network
- PCIe — PCI Express Controller
- SoC — System-on-Chip

Category Definitions

Errata are classified according to system impact and the availability of a workaround.

1. Major impact, no workaround is available. An issue 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. A substantial risk of encountering the same or additional issues, including performance issues, exist after the workaround is implemented.
3. Major impact, workaround available. Application of the workaround either eliminates the issue, or reduces it to a minor impact issue, or results in significant performance degradation.
4. Minor impact, no workaround is available. Minor impact issues result in slight to moderate performance degradation, or are a functional variance from specification.
5. Minor impact, workaround is available. Minor impact issues result in slight to moderate performance degradation, or are a functional variance from specification.
6. Design enhancement.

List of Abbreviations and Acronyms

Table 1: List of Abbreviations and Acronyms Used in the Document (Sheet 1 of 3)

TERM	DESCRIPTION
1P	Single-Socket Platform
2P	Dual-Socket Platform
ASID	Address Space Identifier
ATB	Advanced Trace Bus
BAR	Base Address Register

Table 1: List of Abbreviations and Acronyms Used in the Document (Sheet 2 of 3)

TERM	DESCRIPTION
BHB	Branch History Buffer
BMC	Baseboard Management Controller
BTB	Branch Target Buffer
CA	Completer Abort
CCM	Core Cluster Module
CE	Corrected Error
CFI	Fault Handling Interrupt Corrected Errors
CPLD	Complex Programmable Logic Device
CTO	Completion Timeout
DC IVAC	Data or Unified Cache Line Invalidate by VA
DC ZVA	Data Cache Zero by VA
DE	Deferred Error
ECC	Error Correcting Code
EL2	Exception Level 2
ELR	Exception Link Register
ERRIIDR	Implementation Identification Register
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macrocell
ETR	Embedded Trace Router
FAR	Fault Address Register
FHI	Fault Handling Interrupt
GIC	Generic Interrupt Controller
HPFAR	Hypervisor IPA Fault Address Register
IPA	Intermediate Physical Address
L2C	Level 2 Cache
LDRAA/LDRAB	Load Register, with Pointer Authentication
MIDR	Main ID Register
MMIO	Memory Mapped I/O
NOP	No Operation

Table 1: List of Abbreviations and Acronyms Used in the Document (Sheet 3 of 3)

TERM	DESCRIPTION
PAC	Pointer Authentication Code
PE	Processing Element
PIO	Programmed Input/Output
PMU	Performance Monitoring Unit
RP	Root Port
SBSA	Server Base System Architecture
SECpro	Security Processor
TLP	Transaction Layer Packet
UE	Uncorrected Error
UR	Unsupported Request
VA	Virtual Address
VMID	Virtual Machine Identifier
WFE	Wait For Event
WFI	Wait For Interrupt

Errata Summary

Table 2: Errata Summary (Sheet 1 of 2)

ERRATA NO.	CATEGORY	DESCRIPTION	PAGE
AC03_CPU_12	3	Branch history may allow control of speculative execution across software contexts.	*5*
AC03_CPU_14	5	Timer CVAL programming of a delta greater than 2^{63} will result in incorrect behavior.	*6*
AC03_CPU_16	3	In rare scenarios, a poisoned cache line in an L2C may lose poison state on snoop to another L2C.	*7*
AC03_CPU_29	5	STALL_SLOT_FRONTEND, STALL_FRONTEND, STALL continue counting in WFx state.	*8*
AC03_CPU_36	5	CPU can take an invalid exception, if an asynchronous exception to EL2 occurs while EL2 software is changing the EL2 exception controls.	*9*
AC03_CPU_38	3	Certain bits in the Virtualization Translation Control Register and Translation Control Registers do not follow RES0 semantics.	*10*
AC03_CPU_39	4	HCR_EL2.TIDCP traps extra, UNDEFINED cases.	*11*
AC03_CPU_40	5	Virtual Address (VA) tag multi-way hit in the L1 data cache may result in false reporting of a correctable data error.	*12*

Table 2: Errata Summary (Sheet 2 of 2)

ERRATA NO.	CATEGORY	DESCRIPTION	PAGE
AC03_CPU_41	4	L1D_CACHE_INVAL PMU overcounts in some situations.	*13*
AC03_DEBUG_6	5	ETM memory-mapped accesses are treated as external debugger accesses.	*14*
AC03_DEBUG_10	5	Only 32-bit aligned accesses to 64-bit CoreSight ETM registers are permitted.	*15*
AC03_SOC_1	5	SPI-NOR boot failure due to SECpro ROM accessing SPI-NOR not resetting the SPI-NOR device before accessing.	*16*

AC03_CPU_12: Branch history may allow control of speculative execution across software contexts.

Functional Unit: CPU

Category: 3

Overview:

Spectre BHB is a variant of Spectre-v2 in which malicious code uses the shared branch history (stored in the CPU BHB) to influence mispredicted branches in the victim's hardware context. Speculation caused by these mispredicted branches can be used to infer protected information through cache allocation.

Impact:

Software can use a loop of taken branches to overwrite the branch history on any entry to a higher exception level. This prevents any history from less privileged code from influencing branch target predictions at the new higher privilege level. The same loop that is used for other Arm cores applies to the AmpereOne CPU, as described here.

The loop is as follows:

```
    MOV X0, #K  // PE-specific number
  loop:
    B PC+4
    SUBS x0, x0, #1
    BNE loop
    SB
```

Workaround:

For the AmpereOne CPU, a 'K' value of 11 is sufficient to mitigate Spectre-BHB.

AC03_CPU_14: Timer CVAL programming of a delta greater than 2^{63} will result in incorrect behavior.

Functional Unit: CPU

Category: 5

Overview:

In scenarios where the CompareValue (*CVAL) is greater than or equal to 2^{63} (or 292 years) difference from the counter being compared against, the AmpereOne TimerConditionMet calculation will be wrong (precisely opposite of the expected behavior). This limits Timer functionality to only configure timer interrupts to be within 292 years into the future.

Impact:

There is no expected practical use case for setting a CVAL delta this large. If there is any code that attempts to disable the timer by setting a value > 292 years into the future instead of actually masking it, then it is possible that would result in the timer firing immediately.

Workaround:

Software must enforce that the CVAL value programmed does not exceed a delta of 2^{63} with the counter being compared against.

AC03_CPU_16: In rare scenarios, a poisoned cache line in an L2C may lose poison state on snoop to another L2C.

Functional Unit: CPU

Category: 3

Overview:

In certain rare conditions, while tracking of a poisoned cache line that originated as a data uncorrected error (UE) from another processing element (PE) L2 cache, the poison state (also known as a “data error”) of that cache line may be lost, resulting in the possibility of the cache line to be synchronously consumed without an external abort exception at the PE.

The original UE detection will still be logged and reported to software in the appropriate RAS error node where the UE had originated.

The conditions of this erratum have only been encountered in controlled pre-silicon simulation environments.

Impact:

In this extremely unlikely scenario, the PE will silently consume corrupted data with no synchronous exception. However, this corrupted data was previously detected and reported asynchronously (for example, via Error Recovery Interrupt). On detection of the original error, it would have been reported via error records as an uncorrected error recoverable (UER).

Eventually, software will attempt to offline this memory based on typical memory fault recovery flows triggered via an ACPI Platform Error Interface (APEI) notification.

However, due to the imprecise handling of this error, the cache line may be synchronously consumed without an external abort exception. Therefore, the true severity of this error when these conditions are met would be uncontainable (UC).

Workaround:

Any L2 cache uncorrected error recoverable (UER) severity due to a Data UE must be treated by software as a fatal (uncontainable) error.

AC03_CPU_29: STALL_SLOT_FRONTEND, STALL_FRONTEND, STALL continue counting in WFx state.

Functional Unit: CPU

Category: 5

Overview:

The impacted counters continue counting when the core enters WFx state. Since CPU_CYCLES correctly stops counting in WFx state, this results is a deviation from the Architecture specification, which requires the STALL_* counters to count as a subset of CPU_CYCLES.

Impact:

Accesses to these counters may be incorrect, either through direct register reads, or via the perf tool. TDA metrics reported by perf tool are not impacted – perf tool uses alternate equations to derive metrics (equations are listed in the workaround).

Workaround:

The impacted counters can be calculated as follows:

- STALL_SLOT_FRONTEND[corrected] = (CPU_CYCLES × 4) – OP_SPEC – STALL_SLOT_BACKEND
- STALL_FRONTEND[corrected] =
  STALL_FRONTEND[reported] – ((STALL_SLOT_FRONTEND[reported] – STALL_SLOT_FRONTEND[corrected]) / 4)
- STALL[corrected] = STALL[reported] – (STALL_FRONTEND[reported] – STALL_FRONTEND[corrected])

AC03_CPU_36: CPU can take an invalid exception, if an asynchronous exception to EL2 occurs while EL2 software is changing the EL2 exception controls.

Functional Unit: CPU

Category: 5

Overview:

If an Asynchronous Exception to EL2 occurs, while EL2 software is changing the EL2 exception control bits from a configuration where asynchronous exceptions are routed to EL2 to a configuration where asynchronous exceptions are routed to EL1, the processor may exhibit the incorrect exception behavior of routing an interrupt taken at EL2 to EL1.

The affected system register is HCR_EL2, which contains control bits for routing and enabling of EL2 exceptions.

Impact:

If an Asynchronous Exception (Debug or Interrupt) occurs to EL2, while EL2 software is modifying system register bits that control EL2 exception behavior, the processor may take an exception to an incorrect Exception Level.

Workaround:

EL2 software should prevent Asynchronous Exceptions from being delivered to EL2, while EL2 exception control bits are being changed. This may be done either via masking with PSTATE.DAIF or other mechanisms. In addition, proper synchronization (as required by the ARM ARM, typically ISB) must be performed to ensure completion of the control bit updates, before Asynchronous Exceptions can be allowed to resume.

AC03_CPU_38: Certain bits in the Virtualization Translation Control Register and Translation Control Registers do not follow RES0 semantics.

Functional Unit: CPU

Category: 3

Overview:

ID_AA64_MMFR1_EL1.HAFDBS will report a value of 0b0000 indicating that the hardware update of the access flag and dirty state are not supported. With FEAT_HAFDBS not supported, the register bits in the Virtualization Translation Control Register (VTCR_EL2) and Translation Control Registers (TCR_EL1, TCR_EL2) for enabling/disabling hardware management of access flag and dirty state – specifically VTCR_EL2.{HA, HD}, TCR_EL1.{HA, HD} and TCR_EL2.{HA, HD}, respectively must follow the RES0 semantics. These bits do not follow the RES0 semantics.

Impact:

Setting VTCR_EL2.{HA, HD}, TCR_EL1.{HA, HD} or TCR_EL2.{HA, HD} can lead to unpredictable behavior.

Workaround:

System software/virtualization system software must not set the TCR_EL1.{HA, HD}, TCR_EL2.{HA, HD} or VTCR_EL2.{HA, HD} bits.

AC03_CPU_39: HCR_EL2.TIDCP traps extra, UNDEFINED cases.

Functional Unit: CPU

Category: 4

Overview:

When execution is at EL1 and HCR_EL2.TIDCP is set, UNDEFINED MSR/MRS instructions with op0==2, CRn={11, 15} trap to EL2 instead of causing an UNDEFINED fault.

Impact:

The hypervisor will receive unexpected traps for some undefined system registers.

Workaround:

When HCR_EL2.TIDCP is set and software running at EL2 receives any MSR/MRS traps from EL1 with op0==2, CRn=={11, 15}, an UNDEFINED fault should be injected into EL1.

AC03_CPU_40: Virtual Address (VA) tag multi-way hit in the L1 data cache may result in false reporting of a correctable data error.

Functional Unit: CPU

Category: 5

Affected Version(s): A0, B0

Fixed Version(s): Open

Overview:

The L1D cache has support for both virtual and physical tag arrays. Under certain legal conditions, it is possible that two cache lines are allocated with the same VA tags but different PA tags. A subsequent cache lookup of that VA will see cache hits in two ways of VA tag. This can result in false reporting of a correctable data error in RAS error record 6.

Hits in multiple ways of the VA tag array can also occur due to legitimate correctable data errors. When the VA multi-way hit condition occurs because of a legitimate data error, a VA tag parity error will also be reported in the status of the error record.

When a VA multi-way hit condition occurs, including when not due to a real data error, it is also likely (but not guaranteed) that the RAS hardware incorrectly detects a PA tag array parity error as well.

Impact:

Core Cluster Module (CCM) correctable errors will be reported, logged, and observed by operating system software during intensive workloads that may create the conditions of this erratum.

There is no functional impact when this erratum is encountered.

Workaround:

The error syndrome associated with the non-data-error VA multi-way hit condition is benign and should be ignored (to be differentiated from legitimate VA/PA tag errors). If possible, this error may be filtered by software or firmware. The following is the exact syndrome that must be ignored:

IF ((ERR6STATUS.IERR[2] == 1) && (ERR6STATUS.IERR[1] == 0) && (ERR6STATUS.IERR[4] == 0))

- ERR6STATUS.IERR[2] indicates the VA tag multi-way hit error
- ERR6STATUS.IERR[1] indicates a VA tag parity error
- ERR6STATUS.IERR[4] indicates a parity error in an unrelated structure

When a VA tag multi-way hit error occurs in the absence of a VA tag parity error, then the multi-way hit condition was not due to a legitimate correctable data error and should be ignored. If ERR6STATUS.IERR[4] is also set, this indicates that a (real) parity error in another structure occurred simultaneously with the non-error VA tag array multi-hit condition. In this case, a legitimate error has still occurred. Any bits in the IERR field other than bit 4 should be ignored when (IERR[2] == 1 AND IERR[1] == 0).

Ampere Software Release Package (SRP) revision 3.5.8.1 and beyond will implement filtering of this error condition for impacted products. A new field will be added to ERR6MISC2 for an adjusted corrected error counter (CEA). The CE reporting threshold will no longer solely rely on the CEO value but will be a function of CE Threshold > (CEO – CEA). The CEA field will be incremented each time the ignorable error condition is seen in ERR6STATUS during a reporting window. An NVPARAM will be added to allow disablement of this workaround.

AC03_CPU_41: L1D_CACHE_INVAL PMU overcounts in some situations.

Functional Unit: CPU

Category: 4

Overview:

L1D_CACHE_INVAL PMU counter may overcount L1D cache invalidations by an amount anywhere from 0 up to L2C_DATA_REFILL + L2D_CACHE_INVAL.

Impact:

L1D_CACHE_INVAL PMU may be unreliable for performance tuning.

Workaround:

None.

AC03_DEBUG_6: ETM memory-mapped accesses are treated as external debugger accesses.

Functional Unit: DEBUG

Category: 5

Overview:

The Embedded Trace Macrocell (ETM) provides a trace OS Lock which is used to lock access for exclusive use by an OS. When trace OS Lock is set, certain registers are inaccessible to an external debugger. An external debugger generally clears the trace OS Lock bit when it needs to control these registers, preventing software from accessing them.

Specifically, the “Arm Embedded Trace Macrocell Architecture Specification – ETMv4.0 to ETMv4.6” specifies that memory-mapped accesses to all trace registers, except TRCPRGCTLR, TRCCLAIMCLR, and TRCCLAIMSET, are “OK”, and that external debugger accesses generate an “Error” when trace OS Lock is locked (TRCOSLAR.OSLK=b'1), and are “OK” otherwise, provided that debug power and core power is applied.

On AmpereOne processors, memory-mapped accesses to these registers are treated like external debugger accesses, which generates an “Error” for memory-mapped accesses when trace OS Lock is locked.

Impact:

The trace OS Lock access control mechanism cannot be used to guarantee exclusive memory-mapped access to trace registers by software. Additionally, software, for example the Linux hwtracing/coresight self-hosted trace driver, must ensure the trace OS Lock is unlocked before using memory-mapped accesses to trace registers whose access is controlled by trace OS Lock.

Workaround:

According to the Arm ETM Architecture Specification, when FEAT_TRF is implemented, system instruction access must be implemented, memory-mapped access is deprecated, and whenever possible, software must avoid using deprecated features. Therefore, the recommended workaround is for software to use system instruction accesses instead of memory-mapped accesses when accessing trace registers.

If software does use memory-mapped accesses, software should clear the trace OS Lock bit before initiating memory-mapped accesses to any trace registers except TRCPRGCTLR, TRCCLAIMCLR, and TRCCLAIMSET, and take care to ensure an external debugger is not accessing these registers.

AC03_DEBUG_10: Only 32-bit aligned accesses to 64-bit CoreSight ETM registers are permitted.

Functional Unit: DEBUG

Category: 5

Overview:

64-bit accesses to CoreSight registers are not supported. This impacts 64-bit accesses to CoreSight debug registers, Performance Monitor registers, CTI registers, and ETM registers.

Impact:

Software will encounter a bus error if accessing 64-bit CoreSight registers as a 64-bit access.

Workaround:

Software should always access all CoreSight registers using 32-bit aligned accesses.

AC03_SOC_1: SPI-NOR boot failure due to SECpro ROM accessing SPI-NOR not resetting the SPI-NOR device before accessing.

Functional Unit: SOC

Category: 5

Overview:

On platforms that do not tie the SPI-NOR flash reset pin to the system reset signal (SYS_RESETN), the SPI-NOR flash may be in an unknown state out of a cold reset. The SECpro ROM coming out of reset will immediately attempt to access the SPI-NOR flash device using 3-byte address mode. If the SPI-NOR device has been left in 4-byte address mode, this will result in SPI-NOR command failures. As a result, the SECpro ROM will fail to boot.

This may occur if the system is abruptly reset while the system firmware is accessing the SPI-NOR (for example, during boot or during other SPI-NOR read or write accesses).

Impact:

When entering this system state, the SECpro ROM will not boot until an AC power cycle is performed to ensure the SPI-NOR device is put into a reset state.

Workaround:

1. If the SPI-NOR device has a reset pin, the reset pin may be connected to the system reset to ensure that the SPI-NOR device is always reset along with the SoC.
2. If the SPI-NOR device does not have a reset pin, it is recommended that the platform (for example, BMC or CPLD) intercept the reset and send necessary commands to the SPI-NOR device to put it into the initial reset state.

Revision History

ISSUE	DATE	DESCRIPTION OF MODIFICATIONS
0.80	August 23, 2024	Added erratum AC03_CPU_16 on *page 7*.
		Added erratum AC03_CPU_40 on *page 12*.
		Added erratum AC03_CPU_41 on *page 13*.
		Added erratum AC03_SOC_1 on *page 16*.
0.75	February 13, 2024	Updated erratum AC03_CPU_36 on *page 9*.
		Updated erratum AC03_CPU_39 on *page 11*.
0.70	November 28, 2023	Added erratum AC03_CPU_36 on *page 9*.
0.65	July 27, 2023	Added erratum AC03_CPU_29 on *page 8*.
		Added erratum AC03_CPU_38 on *page 10*.
		Added erratum AC03_CPU_39 on *page 11*.
0.60	January 16, 2023	Added erratum AC03_CPU_12 on *page 5*.
		Added erratum AC03_CPU_14 on *page 6*.
		Added erratum AC03_DEBUG_6 on *page 14*.
		Added erratum AC03_DEBUG_10 on *page 15*.

August 23, 2024

Ampere Computing reserves the right to change or discontinue this product without notice.

While the information contained herein is believed to be accurate, such information is preliminary, and should not be relied upon for accuracy or completeness, and no representations or warranties of accuracy or completeness are made.

The information contained in this document is subject to change or withdrawal at any time without notice and is being provided on an "AS IS" basis without warranty or indemnity of any kind, whether express or implied, including without limitation, the implied warranties of non-infringement, merchantability, or fitness for a particular purpose.

Any products, services, or programs discussed in this document are sold or licensed under Ampere Computing's standard terms and conditions, copies of which may be obtained from your local Ampere Computing representative. Nothing in this document shall operate as an expressed or implied license or indemnity under the intellectual property rights of Ampere Computing or third parties.

Without limiting the generality of the foregoing, any performance data contained in this document was determined in a specific or controlled environment and not submitted to any formal Ampere Computing test. Therefore, the results obtained in other operating environments may vary significantly. Under no circumstances will Ampere Computing be liable for any damages whatsoever arising out of or resulting from any use of the document or the information contained herein.

AMPERE®

Ampere Computing
4655 Great America Parkway, Santa Clara, CA 95054
Phone: (669) 770-3700
https://www.amperecomputing.com

Ampere Computing reserves the right to make changes to its products, its datasheets, or related documentation, without notice and warrants its products solely pursuant to its terms and conditions of sale, only to substantially comply with the latest available datasheet.

Ampere, Ampere Computing, the Ampere Computing and 'A' logos, Altra, and AmpereOne are registered trademarks of Ampere Computing.

Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All other trademarks are the property of their respective holders.