ERRATA LIST

Apollo4 Plus SoC, Apollo4 Blue Plus SoC

Ultra-low Power Apollo SoC Family

Doc. ID: SE-A4P-3p0

Doc. Revision: 3.0, October 2022

Legal Information and Disclaimers

AMBIQ MICRO INTENDS FOR THE CONTENT CONTAINED IN THE DOCUMENT TO BE ACCURATE AND RELIABLE. THIS CONTENT MAY, HOWEVER, CONTAIN TECHNICAL INACCURACIES, TYPOGRAPHICAL ERRORS OR OTHER MISTAKES. AMBIQ MICRO MAY MAKE CORRECTIONS OR OTHER CHANGES TO THIS CONTENT AT ANY TIME. AMBIQ MICRO AND ITS SUPPLIERS RESERVE THE RIGHT TO MAKE CORRECTIONS, MODIFICATIONS, ENHANCEMENTS, IMPROVEMENTS AND OTHER CHANGES TO ITS PRODUCTS, PROGRAMS AND SERVICES AT ANY TIME OR TO DISCONTINUE ANY PRODUCTS, PROGRAMS, OR SERVICES WITHOUT NOTICE.

THE CONTENT IN THIS DOCUMENT IS PROVIDED "AS IS". AMBIQ MICRO AND ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS CONTENT FOR ANY PURPOSE AND DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS CONTENT, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHT.

AMBIQ MICRO DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT OF AMBIQ MICRO COVERING OR RELATING TO THIS CONTENT OR ANY COMBINATION, MACHINE, OR PROCESS TO WHICH THIS CONTENT RELATE OR WITH WHICH THIS CONTENT MAY BE USED.

USE OF THE INFORMATION IN THIS DOCUMENT MAY REQUIRE A LICENSE FROM A THIRD PARTY UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF THAT THIRD PARTY, OR A LICENSE FROM AMBIQ MICRO UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF AMBIQ MICRO.

INFORMATION IN THIS DOCUMENT IS PROVIDED SOLELY TO ENABLE SYSTEM AND SOFTWARE IMPLEMENTERS TO USE AMBIQ MICRO PRODUCTS. THERE ARE NO EXPRESS OR IMPLIED COPYRIGHT LICENSES GRANTED HEREUNDER TO DESIGN OR FABRICATE ANY INTEGRATED CIRCUITS OR INTEGRATED CIRCUITS BASED ON THE INFORMATION IN THIS DOCUMENT. AMBIQ MICRO RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN. AMBIQ MICRO MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE REGARDING THE SUITABILITY OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, NOR DOES AMBIQ MICRO ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT, AND SPECIFICALLY DISCLAIMS ANY AND ALL LIABILITY, INCLUDING WITHOUT LIMITATION CONSEQUENTIAL OR INCIDENTAL DAMAGES. “TYPICAL” PARAMETERS WHICH MAY BE PROVIDED IN AMBIQ MICRO DATA SHEETS AND/OR SPECIFICATIONS CAN AND DO VARY IN DIFFERENT APPLICATIONS AND ACTUAL PERFORMANCE MAY VARY OVER TIME. ALL OPERATING PARAMETERS, INCLUDING “TYPICALS” MUST BE VALIDATED FOR EACH CUSTOMER APPLICATION BY CUSTOMER’S TECHNICAL EXPERTS. AMBIQ MICRO DOES NOT CONVEY ANY LICENSE UNDER NEITHER ITS PATENT RIGHTS NOR THE RIGHTS OF OTHERS. AMBIQ MICRO PRODUCTS ARE NOT DESIGNED, INTENDED, OR AUTHORIZED FOR USE AS COMPONENTS IN SYSTEMS INTENDED FOR SURGICAL IMPLANT INTO THE BODY, OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, OR FOR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE AMBIQ MICRO PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR. SHOULD BUYER PURCHASE OR USE AMBIQ MICRO PRODUCTS FOR ANY SUCH UNINTENDED OR UNAUTHORIZED APPLICATION, BUYER SHALL INDEMNIFY AND HOLD AMBIQ MICRO AND ITS OFFICERS, EMPLOYEES, SUBSIDIARIES, AFFILIATES, AND DISTRIBUTORS HARMLESS AGAINST ALL CLAIMS, COSTS, DAMAGES, AND EXPENSES, AND REASONABLE ATTORNEY FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PERSONAL INJURY OR DEATH ASSOCIATED WITH SUCH UNINTENDED OR UNAUTHORIZED USE, EVEN IF SUCH CLAIM ALLEGES THAT AMBIQ MICRO WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE PART.

Table of Content

Silicon Errata for the **Apollo4 Plus SoC**

1. Introduction

This document is a detailed compilation of known device errata for the general availability revision of the Apollo4 Plus SoC and Apollo4 Blue Plus SoC. Unless stated otherwise, all listed errata apply to both Blue and non-Blue versions, and all packages, of the SoC.

2. Document Revision History

Table 1: Document Revision History

Rev No.	Date	Description
1.0	Jan 2022	Initial General Release Version
2.0	Jun 2022	Updated errata: - ERR063: Removed as fixed. - ERR065: Updated title and application impact. - ERR091: Updated with expanded workaround. Added errata: -ERR099, ERR101 - ERR106, ERR109
3.0	Sep 2022	Added errata: - ERR110 - ERR113

3. Errata Summary List

Below is a list of the errata described in this document. The reference number for each erratum is listed along with its description and link to the page where detailed information can be found.

Reference to fixes on earlier Apollo4 SoC versions implies that those fixes are on the latest Apollo4 Plus SoC revision as well unless otherwise stated.

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
“ERR002: ADC: DMA ISR cannot be properly triggered” on page 9	All existing	No Fix planned	Software workaround
“ERR003: CACHE: Some modes are unsupported” on page 10	All existing	Partial fix implemented (B0)	No workaround
“ERR008: GPIO: Dual-edge interrupts are not vectoring” on page 11	All existing	No fix planned	Software and/or hardware workaround
“ERR038: IOM: CQ fails to complete DMA read transfers” on page 12	All existing	No fix planned	Software workaround
“ERR039: IOS: MISO line is not tri-stated when CE driven high” on page 13	All existing	No fix planned	Software/hardware workaround
“ERR040: IOS: FIFO read gets stuck/stalled” on page 14	All existing	Partially fixed (B0); no further fix planned.	No workaround
“ERR041: USB: Induced D+ output pulse may cause unintended disconnect” on page 15	All existing	No fix planned	Software/hardware workaround
“ERR043: DMA: Incorrect reads/writes near memory boundaries” on page 18	All existing	No fix planned	Software workaround
“ERR046: GPIO: FIEN/FOEN not operational on GPIO0” on page 19	All existing	No fix planned	Software/hardware workaround
“ERR056: USB: High leakage current in USB PHY” on page 20	All existing	No fix planned	Hardware workaround
“ERR064: IOM: I2C Power Save/Restore Failure” on page 21	All existing	No fix planned	Software workaround
“ERR065: MSPI: CM4 hard fault not triggered when it should be” on page 22	All existing	Fix planned for future revision	Software workaround
“ERR066: IOS: Fails in FIFO mode at a lower range of clock frequencies” on page 23	All existing	Fix planned for future revision	No workaround
“ERR071: INFO: Reads of INFO space fail” on page 24	All existing	Fix planned for future revision	Software workaround
“ERR073: GPU: Blit overruns destination texture” on page 25	All existing	Fix planned for future revision	Software workaround
“ERR074: DSI: Image displayed outside the set region” on page 26	All existing	Fix planned for future revision	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
“ERR075: I2S: Clocks incorrectly gated while in deep sleep” on page 27	All existing	Fix planned for future revision	Limited software work-around
“ERR077: DSI: Cannot free DBI when reading DSI register” on page 28	All existing	No fix planned	Software workaround
“ERR078: MSPI: Potential race condition when using RXNEG and RXDQSDELAY concurrently” on page 29	All existing	Fix planned for future revision	Software workaround
“ERR079: INFO0: SIMO Buck cannot be enabled via INFO0 setting” on page 30	All existing	No fix planned	Software workaround
“ERR080: MSPI: Command Queue may disable DMAEN while data is still pending in internal buffer” on page 31	All existing	Fix planned for future revision	Software workaround
“ERR082: Memory: Potential lockup when DC/GFX accesses MSPI/ Extended Memory while MSPI is DMAing to there” on page 32	All existing	No fix planned	System configuration
“ERR085: IOS: Possible failure in FIFO mode in Wrap configuration” on page 33	All existing	No fix planned	Software workaround
“ERR087: MCU_CTRL: POR failure due to VDDC/ VDDF not rising to proper level” on page 34	All existing	No fix planned	Hardware workaround
“ERR088: SDIO: DDR modes not supported” on page 35	All existing	No fix planned	No workaround
“ERR090: ADC: No CNVCMP interrupt for first single scan” on page 36	All existing	No fix planned	Software workaround
“ERR091: ADC: Loss of first scan data” on page 37	All existing	No fix planned	Software workaround
“ERR092: ADC/AUDADC: DMACPL interrupt sometimes fails” on page 38	All existing	No fix planned	Software workaround
“ERR094: BootROM: Does not accept source/destination address higher than 0x101D7FFF” on page 39	All existing	No fix planned	Software workaround
“ERR096: DC: DPI-2 interface is not supported” on page 40	All existing	No fix planned	Software workaround
“ERR097: MSPI: Non-DQS SDR Octal not reliably operable at 48 MHz or 96 MHz” on page 41	All existing	No fix planned	Software workaround
“ERR098: STIMER: Constraints on writing to SCMPRn registers and handling Compare interrupts” on page 42	All existing	No fix planned	Software workaround
“ERR099: IOM: CQ does not pause via the BLE module” on page 44	All existing	No fix planned	No workaround
“ERR101: IOM: Command write causes CQ operations to pause and never restart” on page 45	All existing	No fix planned	Software workaround
“ERR102: IOM: CQ does not pause immediately after triggering event” on page 46	All existing	No fix planned	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
“ERR103: IOM: FIFO threshold interrupt incorrectly triggered” on page 47	All existing	No fix planned	Software workaround
“ERR104: IOM: Data corrupted on I2C when OFFSETCNT=0 and I2CLSB=1” on page 48	All existing	No fix planned	Software workaround
“ERR105: GPU: System hang when GPU fetches data from internal SRAM” on page 49	All existing	No fix planned	Software workaround
“ERR106: AUDADC: MCU hangs when attempting to configure with disabled XTALHS clock” on page 50	All existing	No fix planned	Software workaround
“ERR109: MSPI: Dedicated DQS falling edge delay settings do not work in DEV0DDR register” on page 52	All existing	No fix planned	Software workaround
“ERR110: DAXI: Out-of-order SSRAM read and write returns incorrect read value” on page 53	All existing	No fix planned	Software workaround
“ERR111: MSPI: Delayed MSPI write b-response may cause MSPI state machine deadlock” on page 55	All existing	No fix planned	Software workaround
“ERR112: ADC: Dummy trigger causes immediate (invalid) interrupt” on page 57	All existing	No fix planned	Software workaround
“ERR113: ADC: Occasional corrupt conversion results at 48 MHz” on page 58	All existing	No fix planned	Software workaround

4. Detailed Silicon Errata

This section gives detailed information about each erratum. Information covered for each erratum includes the following:

- **Erratum Reference Number and Title** – Lists reference number and title of the erratum
- **Description** – Provides a detailed description of the erratum
- **Affected Silicon Revisions** – Specifies the silicon revisions on which the erratum exists
- **Application Impact** – Describes the impact of the erratum on a user application
- **Workarounds** – Proposes software or hardware workarounds to minimize or eliminate the risk of the erratum occurring
- **Erratum Resolution Status** – Specifies which silicon revision, if any, that the erratum was initially fixed
- **AmbiqSuite Workaround Status** – Specifies whether the erratum has been worked around in the AmbiqSuite software

4.1 ERR002: ADC: DMA ISR cannot be properly triggered

4.1.1 Description

The ADC DMA ISR cannot be properly triggered and this will result in missing interrupt conditions for the ADC DMA complete and ADC DMA ERROR interrupts.

4.1.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.1.3 Application Impact

Application needs to service more frequent ADC interrupts in order to check for DMA completion following completion of a conversion, scan, or FIFO threshold trigger.

4.1.4 Workarounds

A workaround for this limitation is to use 75% threshold for FIFO fullness and inside the ISR, read DMA_STAT until it is DMA_COMPLETE.

4.1.5 Erratum Resolution Status

This erratum is intended to be fixed in a future silicon revision.

4.1.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

4.2 ERR003: CACHE: Some modes are unsupported

4.2.1 Description

Of the 6 major MRAM cache modes set in CPU_CACHECFG_CONFIG:

- W1_128B_512E = 0x4, // Direct mapped, 128-bit linesize, 512 entries (4 SRAMs active)
- W2_128B_512E = 0x5, // Two-way set associative, 128-bit linesize, 512 entries (8 SRAMs active)
- W1_128B_1024E = 0x8, // Direct mapped, 128-bit linesize, 1024 entries (8 SRAMs active)
- W1_128B_2048E = 0xC, // Direct mapped, 128-bit linesize, 2048 entries (4 SRAMs active)
- W2_128B_2048E = 0xD, // Two-way set associative, 128-bit linesize, 2048 entries (8 SRAMs active)
- W1_128B_4096E = 0xE // Direct mapped, 128-bit linesize, 4096 entries (8 SRAMs active)

Only 4 are supported:

- W1_128B_512E
- W2_128B_512E
- W1_128B_1024E
- W1_128B_4096E

And only one of these modes, W1_128B_4096E, may be used with MSPI memory-mapped/XIP.

4.2.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.2.3 Application Impact

The cache must be configured as one of the supported modes. There is no negative impact to user applications if system can effectively utilize one of the four supported cache configurations. If there is a need to support XIP caching with a 2-way set associative configuration, then there are limitations on the affected revisions to support this.

4.2.4 Workarounds

There is no workaround for this limitation.

4.2.5 Erratum Resolution Status

Support for W2_128B_2048E was added in silicon revisions B0 and exist on all following revisions.

4.2.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

4.3 ERR008: GPIO: Dual-edge interrupts are not vectoring

4.3.1 Description

GPIO dual-edge interrupts are not vectoring to the respective IRQ. Basically for PINCFGn_IRPTENn = 3 (INTANY) the ISR is only called for the falling edges and misses the rising edge. When the individual edges are specified (i.e., IRPTEN =1 or 2), the ISR is called appropriately.

4.3.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.3.3 Application Impact

Dual-edge interrupt capability is not possible without adding additional functionality in the ISR or a hardware modification to utilize a separate GPIO for each edge interrupt.

4.3.4 Workarounds

Two possible workarounds are available to enable both rising and falling edge GPIO interrupts:

1. Initially set IRPTEN to either 1 or 2. When the ISR is called, toggle the setting and handle the interrupt.
2. Physically tie the target pin to an unused GPIO. Set one GPIO for rising edge and the other for falling edge.

4.3.5 Erratum Resolution Status

There is no plan to fix this erratum.

4.3.6 AmbiqSuite Workaround Status

Both of the workarounds cited have been tested in the Ambiqsuite SDK using existing functions.

4.4 ERR038: IOM: CQ fails to complete DMA read transfers

4.4.1 Description

IOM Command Queue (CQ) with DMA fails to transfer the last 1 to 3 bytes left in the read FIFO when the read threshold (FIFORTHR) is set to 4. The CQ doesn't wait until the DMA completes for the specified TOTCOUNT. The CQ ends the transfer before the DMA completes. Due to this, the residual bytes are not transferred by DMA.

4.4.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.4.3 Application Impact

If an IOM transfer count is not a multiple of 4, then a DMA transfer may not complete. Depending on what is being transferred, and by which peripheral, an incomplete transfer could cause various anomalies within the system.

4.4.4 Workarounds

A workaround is to use a transfer count which is a multiple of 4, or a read threshold of 8 or more.

4.4.5 Erratum Resolution Status

There are no plans at this time to fix this erratum.

4.4.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK offers HAL functionality to implement the workaround described above.

4.5 ERR039: IOS: MISO line is not tri-stated when CE driven high

4.5.1 Description

When configured as a SPI slave using the IOS module, the Apollo4 does not tri-state the MISO pin when CE is driven high. Instead, the MISO pin is driven static low when CE is driven high.

4.5.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.5.3 Application Impact

If there are multiple slaves on the same SPI bus as an Apollo4 configured as a SPI slave, other slave devices will be prevented from driving data onto the MISO line.

4.5.4 Workarounds

Workarounds include:

1. Do not have any other slave devices on the SPI bus.
2. Add an external tri-state buffer between the Apollo4 MISO pin and the MISO line of the SPI bus so that the MISO pin on Apollo4 does not drive the bus when CE is driven high by the SPI master device.
3. Implement a software workaround that reconfigures and tri-states the Apollo4 MISO pin when the Apollo4 CE signal is driven high by the SPI master device.

4.5.5 Erratum Resolution Status

There are no plans at this time to fix this erratum.

4.5.6 AmbiqSuite Workaround Status

There is no specific workaround in the AmbiqSuite SDK for this issue.

4.6 ERR040: IOS: FIFO read gets stuck/stalled

4.6.1 Description

If the SPI master pauses the SPI SCK, an Apollo4 slave CPU may get stuck waiting for the next SCK from the SPI master when accessing IOS registers. The control state machine of the IOS assumes that once the interface starts an operation (read or write), it finishes it and the bus is held off until that happens because only one operation can take place with the LRAM at a time. The read or write request is asserted for one interface clock cycle, so if the clock stops the request will be held and the IOS (and MCU) will be stalled.

This could also happen when inside an ISR, causing extended delays in interrupt context. For example, the AmbiqSuite SDK HAL implements larger size IOS nonblocking transactions using a SW assisted replenishing of the hardware FIFO, by servicing the FSIZE interrupts. This issue could cause the ISR handler to get stuck when servicing the interrupt.

4.6.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.6.3 Application Impact

Getting stuck in an ISR may block other interrupts from being serviced. For example, such an occurrence may be the STIMER tick interrupt being blocked resulting in delayed scheduling of the RTOS. After the occurrence of several such blocking events, the STIMER may get overrun (e.g., the compare value is less than the counter without INSTAT getting set).

4.6.4 Workarounds

There is no workaround for this issue. The master needs to ensure that it does not insert long pauses in the clock once it has started a transaction.

4.6.5 Erratum Resolution Status

This erratum was partially fixed in silicon rev. B0. Operates at 16 MHz, 1 MHz, 400 kHz, and 100 kHz, but fails at frequencies higher than 1 MHz and less than 16 MHz due to this issue.

4.6.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

4.7 ERR041: USB: Induced D+ output pulse may cause unintended disconnect

4.7.1 Description

An output pulse on the D+ line while VDDUSB0P9 and/or VDDUSB33 are/is rising may be interpreted by a host as a connect event immediately followed by the disconnect one, causing the host USB SW to report about the unexpected disconnect from Apollo4. Such a report/message could be safely ignored for the USB-compliant hosts which would attempt USB device re-enumeration. An enumeration attempt performed when all USB power rails are stable succeeds.

4.7.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.7.3 Application Impact

The Apollo4 may fail to connect or stay connected with USB hosts that deviate from strict USB 2.0 specifications.

4.7.4 Workarounds

USB 2.0 compliant hosts are not affected by this erratum. While there is no workaround for the hosts that are not strictly USB 2.0 compliant, system initialization and shutdown sequences listed below minimize the chances of the non-compliant host being affected by the erratum.

NOTE: These USB power-up/power-down sequences will be (or is) updated in the Apollo4 Datasheet version 0.7.2 from what was recommended in version 0.7.1 and previous versions.

Silicon Revision A Start-up Sequence:

1. Enable power to VDDUSB33.
2. Enable Apollo 4 internal power rail to USB interface (VDDF_USB_SW).
3. Enable power to VDDUSB0P9.
4. Initialize the USB Device Controller, USB PHY, and USB Host Stack.
5. Enable host connect (call am_hal_usb_attach()).

Silicon Revision B Start-up Sequence:

1. Enable Apollo 4 internal power rail to USB interface (VDDF_USB_SW).
2. Clear USB PHY reset bits in MCU control registers - this forces USB PHY resets.
   - Assert USB PHY POR reset by clearing the MCUCTRL_USBPHYRESET_USBPHYPORRSTDIS bit.
   - Assert USB PHY UTMI reset by clearing the MCUCTRL_USBPHYRESET_USBPHYUTMIRSTDIS bit.

Table 3: MCUCTRL_USBPHYRESET Register

MCUCTRL_USBPHYRESET Register Address: 0x40020418			
Bit No.	**Name**	**R/W**	**Description**
1	USBPHYUTMIRSTDIS	W	De-assert USB PHY UTMI reset
0	USBPHYPORRSTDIS	W	De-assert USB PHY POR reset

3. Power on the USB digital domain (USB controller and the digital core of USB PHY) after the external power supply 3.3V to USB PHY is stable.
4. Power on USB PHY analog 0.9V power immediately after powering on the USB digital domain or simultaneously with the digital domain.
5. Perform battery charger (BC) detection (if required).
6. Disable BC detection voltage sources, current sources, pull-ups, and pull-downs; disconnect BC detection comparators. This step shall be performed regardless of performing BC detection because, by POR default state, DP/DM lines are pulled down for floating input detection.
7. Set USB PHY reset disables in the MCU Control USBPHYRESET register. Setting these removes the forced USB PHY reset forces and allows USB controller to manage them.
   - De-assert USB PHY POR reset by setting the MCUCTRL_USBPHYRESET_USBPHYPORRSTDIS bit.
   - De-assert USB PHY UTMI reset by setting the MCUCTRL_USBPHYRESET_USBPHYUTMIRSTDIS bit.
8. Enable host connect. (call am_hal_usb_attach()).

On a USB VBUS disconnect interrupt, MCU should perform the following actions:

Silicon Revision A Shut-down Sequence:

1. Put USB device in SUSPEND state.
2. Disable host connect (call am_hal_usb_detach()).
3. Disable power to USB PHY external rail(s).
4. Disable Apollo 4 internal power rail to USB interface (VDDF_USB_SW).

Silicon Revision B Shut-down Sequence:

1. Put USB device in SUSPEND state.
2. Disable host connect (call am_hal_usb_detach()).
3. Clear USB PHY reset bits in MCU control registers - this forces USB PHY resets.
   - Assert USB PHY POR reset by clearing the MCUCTRL_USBPHYRESET_USBPHYPORRSTDIS bit.
   - Assert USB PHY UTMI reset by clearing the MCUCTRL_USBPHYRESET_USBPHYUTMIRSTDIS bit.
4. Disable power to USB PHY external rail(s).
5. Disable Apollo 4 internal power rail to USB interface (VDDF_USB_SW).

4.7.5 Erratum Resolution Status

There are no plans at this time to fix this erratum.

4.7.6 AmbiqSuite Workaround Status

No workaround is needed in the AmbiqSuite SDK.

4.8 ERR043: DMA: Incorrect reads/writes near memory boundaries

4.8.1 Description

The APBDMA reads or writes unexpected values near the boundaries of target memories when the target DMA start address is not 32-byte aligned. This issue may occur when DMA transactions straddle SRAM and SSRAM, SSRAM and DSP0_IRAM, DSP0_IRAM and DSP1_DRAM, or DSP1_DRAM and DSP1_IRAM.

The Display Controller also exhibits a similar issue when the frame buffer is set across memories (SSRAM - DSP0_RAM, DSP0_RAM - DSP1_RAM, DSP_IRAM - DSP_DRAM) and starts on addresses which are not 32-byte aligned and even some that are 32-byte aligned addresses. Frame buffer start addresses must be *64-byte* aligned addresses.

4.8.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.8.3 Application Impact

DMA accesses can result in erroneous data when a non-aligned read/write crossing a memory boundary is made.

4.8.4 Workarounds

The workaround for this issue is to ensure that the DMA start addresses are 32-byte aligned for the APB-DMA, and 64-byte aligned for the Display Controller.

4.8.5 Erratum Resolution Status

There are no plans at this time to fix this erratum.

4.8.6 AmbiqSuite Workaround Status

No software workaround is needed in the AmbiqSuite SDK.

4.9 ERR046: GPIO: FIEN/FOEN not operational on GPIO0

4.9.1 Description

Function selections Force Input Enable Active (FIEN) and Force Output Enable Active (FOEN) are not operational on GPIO0. All other selectable functions for GPIO0 work as documented, and FIEN/FOEN operations on other GPIO are not affected by this erratum.

4.9.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.9.3 Application Impact

User applications may be affected by this erratum if the FIEN or FOEN function is needed on GPIO0. When either the FIEN and FOEN bit is set, input or output enable is active regardless of the function selected for the GPIO and when that function sets the enable. Therefore the selected function enables the input/output only when needed.

4.9.4 Workarounds

The workaround for this limitation is to use a GPIO other than GPIO0, if possible.

4.9.5 Erratum Resolution Status

There are no plans at this time to fix this erratum.

4.9.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK related to this loss of functionality on GPIO0.

4.10 ERR056: USB: High leakage current in USB PHY

4.10.1 Description

There is a high-current leakage path drawing about 34 mA from the 3.3 V VDDUSB33 rail when the digital PHY rail (0.8 V) is not powered.

4.10.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.10.3 Application Impact

This issue affects user applications by drawing excessive power under the conditions described above.

4.10.4 Workarounds

A workaround for this issue is to add a weak pull-down resistor (~10 Mohm) on both D+ and D-.

4.10.5 Erratum Resolution Status

There are no plans at this time to fix this erratum.

4.10.6 AmbiqSuite Workaround Status

There is no software workaround for this issue in the AmbiqSuite SDK.

4.11 ERR064: IOM: I2C Power Save/Restore Failure

4.11.1 Description

In internal testing, a power save and restore failure occurs at both 1.755 V and 2.2 V. The power save returns status 3 (AM_HAL_STATUS_IN_USE) and power restore returns status 7 (AM_HAL_STATUS_INVALID_OPERATION). Failures seem to only occur during the first iteration after the frequency has changed.

When the compile switch macro is set to IOM_TEST_NO_POWER_SAVE_RESTORE, all tests for I2C passes without issue. Debug shows that the am_hal_iom_power_ctrl() function quits with error AM_HAL_STATUS_IN_USE. Adding debug code in the HAL shows that the STATUS register read returns a value 0 (unknown) instead of 4 (IDLE).

4.11.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.11.3 Application Impact

User applications may fail when using I2C power save and power restore functions without proper I2C configuration to prevent the failure.

4.11.4 Workarounds

A workaround for this issue is to set up the I2C as follows:

1. Update 1 MHz I2C initialization to include setting MI2CCFG_SMPCNT = 2. This sets the number of base clock cycles to wait before sampling the SCL clock to determine if a clock stretch event has occurred.
2. Update power save/restore operations to sequence the SUBMODCTL manipulation as a safety measure.

4.11.5 Erratum Resolution Status

There are no plans at this time to fix this erratum.

4.11.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK release on 8Feb2021 includes changes in the IOM HAL that prevents this issue from occurring.

.

4.12 ERR065: MSPI: CM4 hard fault not triggered when it should be

4.12.1 Description

The CM4 hangs waiting for a ready signal from an MSPI instance when the MSPI memories are read before powering up MSPI. The CM4 should raise a hard fault as it does for every other memory (SSRAM, Extended RAMs etc.). When MSPI is powered down, it is still in reset (hreset_mspi_n) and can't send a valid response to the AXI subsystem. Other memories are reset by different reset logic which is not dependent on the memory being powered on or off.

When the issue happens, the CPU and buses seem to hang where none of the APB/PPB/AHB addresses are accessible anymore. This occurs for MSPI memory of all three MSPI instances. Similar access on powered-down SRAM (0x10002000), shared SRAM (0x10060000) and extended memory (0x10160000) triggers the expected hardfault.

4.12.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.12.3 Application Impact

This issue may hang user applications which access an XIP address through a powered-down MSPI.

4.12.4 Workarounds

The workaround for this issue is to ensure that MSPI memories are not read before powering up MSPI.

4.12.5 Erratum Resolution Status

This erratum is intended to be fixed in a future silicon revision.

4.12.6 AmbiqSuite Workaround Status

No workaround in the SDK for this issue.

4.13 ERR066: IOS: Fails in FIFO mode at a lower range of clock frequencies

4.13.1 Description

A modification which fixed the clock stopping problem in silicon revision B0 created a timing window in the IOS operating in FIFO mode where the SPI interface receives incorrect data during a burst write operation. This has been seen to occur also when the CPU is accessing the LRAM at the same time as a master write. Failures only occur at interface frequencies between ~500 kHz and ~15 MHz.

4.13.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.13.3 Application Impact

This issue limits the rate that a master clocks the IOSLAVE in FIFO mode.

4.13.4 Workarounds

1. Clock the IOS in SPI or I2C mode at 15 MHz or higher (SPI only) or 500 kHz or lower (SPI or I2C).

2. This imposes a requirement on the external Master device, which must conform to these frequency requirements.

4.13.5 Erratum Resolution Status

This erratum is intended to be fixed in a future silicon revision.

4.13.6 AmbiqSuite Workaround Status

There is no workaround in the SDK for this issue.The master frequency is completely controlled by the Master device, so there are no software implications for AmbiqSuite.

4.14 ERR071: INFO: Reads of INFO space fail

4.14.1 Description

Accesses to INFO spaces fail due to a missing latch on the IFREN signal in the TMC MRAM controller. Allowing IFREN to drop in the middle of transaction before any INFO read or write transactions are complete will result in undefined behavior.

4.14.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.14.3 Application Impact

This issue affects user applications by making INFO0 and INFO1 accesses unreliable.

4.14.4 Workarounds

Temporarily program the TMC MRAM controller to use a Tcycrd of 1 (2 cycles) to allow IFREN to stay asserted one cycle longer before any info space access. Then revert Tcycrd to 0 when done with info space accesses. Restore remainder of the TMC MRAM controller r_timer1 (which contains Tcycrd) to reset defaults after SBR/SBL but before initial deep sleep entry.

4.14.5 Erratum Resolution Status

This erratum is intended to be fixed in a future silicon revision.

4.14.6 AmbiqSuite Workaround Status

The above-mentioned Tcycrd-based workaround running from TCM/SRAM has NOT been included as part of the MRAM HAL functions in the AmbiqSuite SDK for an INFO space read.

4.15 ERR073: GPU: Blit overruns destination texture

4.15.1 Description

A blit can overrun the destination texture (framebuffer) and can write beyond the clipping height boundary and texture dimensions. There is a bug in the GPU hardware which occurs when the right/bottom edge of a geometry is in the range (RESX+0.5, RESY+1).

4.15.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.15.3 Application Impact

This issue affects blit display quality in user applications.

4.15.4 Workarounds

In order to not write outside the framebuffer, a slightly larger framebuffer should be allocated, e.g., (RESX+1, RESY+1).

4.15.5 Erratum Resolution Status

This erratum is intended to be fixed in a future silicon revision.

4.15.6 AmbiqSuite Workaround Status

There is no software workaround needed for this issue in the AmbiqSuite SDK. The stated workaround can be implemented in user code to avoid this issue.

4.16 ERR074: DSI: Image displayed outside the set region

4.16.1 Description

An image with artifacts displayed out of the selected region is observed. This is caused by occasional incorrect bytes in region setting commands. This issue occurs with both HS command and LP command.

The root cause for occasional incorrect bytes is that the command packet is formed by multiple writes to DBICMD register. The register bit DBICFG.SPI_HOLD is set to accumulate parameters, and is cleared to send command over the DBI interface. But this clearing is asynchronous and a potential cause for corruption (CDC).

4.16.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.16.3 Application Impact

This issue affects user applications by presenting images incorrectly on the display.

4.16.4 Workarounds

A workaround for this issue is to follow the sequence below to form a command.

1. Delay 20 µs.
2. Stop the display clock.
3. De-assert SPI_HOLD to form command.
4. Enable the display clock.
5. Delay 10 µs.

4.16.5 Erratum Resolution Status

This erratum is intended to be fixed in a future silicon revision.

4.16.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides a software workaround for this issue with the following code:

```
am_util_delay_us(20);
uint32_t ui32Val6 = AM_REGVAL(0x40004084);
AM_REGVAL(0x40004084) = ui32Val6 & 0xffffffcf;
nemadc_MIPI_CFG_out(ui32Mode);                    // de-assert SPI hold
AM_REGVAL(0x40004084) = ui32Val6;
am_util_delay_us(10);
```

4.17 ERR075: I2S: Clocks incorrectly gated while in deep sleep

4.17.1 Description

I2S Master DMA does not work after entering deep sleep. There is no problem when entering normal sleep. When the issue occurs, there are still I2S CLK & WS signals, but no I2S data and no DMA interrupt, and there will only be 64 samples, one TX FIFO size, transferred on the I2S bus, indicating that the DMA is not triggered or not operating correctly.

4.17.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.17.3 Application Impact

This issue prevents user applications from using I2S in deep sleep mode, thereby possibly causing significantly higher power draw.

4.17.4 Workarounds

The workaround is to keep from entering deep sleep when I2S is powered up and expected to operate.

4.17.5 Erratum Resolution Status

This erratum is intended to be fixed in a future silicon revision.

4.17.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK cannot provide a software workaround for this issue.

4.18 ERR077: DSI: Cannot free DBI when reading DSI register

4.18.1 Description

Cannot free DBI when reading register from DSI display, and display is not connected. When executing a DSI read command, the correct sequence is:

1. DBI sends read command.
2. DSI IP translates this command to DSI and sends this command to display.
3. When DSI gets returned data from display, it sets DBI_DataValid and returns data to DBI.
4. DBI receives data and enters idle state.

But when the display is not connected or is damaged, the DSI cannot get returned data and ACK from the display. It will enter error state, and will not set DBI_DataValid. Then the DBI cannot be freed and it will stay in the busy state. The DC status register will indicate "DBI/SPI CS is busy" (bit 14 at 0x400A00FC).

Disabling/enabling DBIB (bit 4 in MODE register) to force DBIB to idle state (bit 26 in DBIB_CFG register) and force CSX to high/low level (bit 29 and bit 30 in DBIB_CFG register), fails to clear bit 14 in NEMADC_REG_STATUS. So the DBIB bus cannot be freed after a read failure.

If MIPICFG_EN_DVALID signal is used, then DBI_DV (DBI_DataValid) signal must also be used to terminate the read transaction.

4.18.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.18.3 Application Impact

This issue may cause the DBI to remain in a busy state when a display panel becomes unconnected or is otherwise unresponsive in a user applications.

4.18.4 Workarounds

When disabling EN_DVALID after read operations in all APIs for DSI reads, the DBI/DSI can be released from busy mode.

4.18.5 Erratum Resolution Status

There are no plans at this time to fix this erratum.

4.18.6 AmbiqSuite Workaround Status

EN_DVALID has been disabled after read operation in all APIs for DSI reads. There is a timeout in the SDK when checking to see if the panel is connected and responding so that DBI/DSI can be released from busy mode.

4.19 ERR078: MSPI: Potential race condition when using RXNEG and RXDQSDELAY concurrently

4.19.1 Description

When (RXNEG + RXDQS) delay combination exceeds 10.4 ns, the byte counter signal for MSPI increments early, causing all receive data on MSPI to be offset by 1 byte. This results in 1 byte of erroneous data at the beginning of a transaction. This applies to non-DQS mode only, and will be seen when tuning MSPI receive data timing.

4.19.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.19.3 Application Impact

This issue has little effect on user applications, as the timing tuning tool takes appropriate action when the race condition is encountered and returns proper delay parameters.

4.19.4 Workarounds

The workaround for this issue is, for > 10.4ns delay, increment the TURNAROUND for 96 MHz (DDR 48 MHz), or use RXCAP for frequencies lower than 96 MHz. Please see the AmbiqSuite Workaround Status below which describes an SDK example utilizing a tuning procedure that discards failing MSPI tuning settings, including failures of this error type.

4.19.5 Erratum Resolution Status

This erratum is intended to be fixed in a future silicon revision.

4.19.6 AmbiqSuite Workaround Status

The Ambiqsuite SDK provides an updated HAL device driver as well as an example program, mspi_ddr_octal_psram_example, which demonstrates how timing sweeps are performed. To work around this issue, the tuning procedure discards any tuning values that results in an error.

4.20 ERR079: INFO0: SIMO Buck cannot be enabled via INFO0 setting

4.20.1 Description

The SIMO buck cannot be enabled via code execution in INFO0. The INFO0 field for SIMOBUCK enable has been deprecated, and MUST be kept as 0.

4.20.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.20.3 Application Impact

This issue has little effects on user applications. The only impact is that the SIMO Buck needs to be initiated in software during system initialization.

4.20.4 Workarounds

Enabling SIMO buck can be done in software.

4.20.5 Erratum Resolution Status

There are no plans at this time to fix this erratum.

4.20.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK workaround for this issue is to enable the SIMO buck via a HAL function: am_hal_pwrctrl_control(AM_HAL_PWRCTRL_CONTROL_SIMOBUCK_INIT, 0).

4.21 ERR080: MSPI: Command Queue may disable DMAEN while data is still pending in internal buffer

4.21.1 Description

The issue occurs when the Command Queue (CQ) has fetched all the data in the MSPI internal buffers and assumes the DMA is complete. The data in the buffer is not flushed to APMEM due to (1) DEV0BOUNDARY_DMATIMELIMIT0 or a DEV0BOUNDARY_DMABOUND0 break is exceeded, or (2) an XIP transaction from GPU or MCU occurs.

When the MSPI XIP DMA finishes per condition (2), there is a transition time of a few cycles that the CQ state machine thinks the data is flushed to APMEM.

The Command Queue state machine writes to the DMAEN register to shut down the DMA. The disabling of DMAEN causes the MSPI XIP DMA's finite state machine (FSM) to hang.

4.21.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.21.3 Application Impact

This issue may affect user applications by losing data in the MSPI buffer by not emptying the buffer before disabling DMA.

4.21.4 Workarounds

To work around this issue, do not disable DMAEN before all data is flushed to external memory. Wait until the following condition is true before all data is flushed to external memory.

(dma_active) | ((CQMASK & CQFLAGS)!=CQMASK).

4.21.5 Erratum Resolution Status

This erratum is intended to be fixed in a future silicon revision.

4.21.6 AmbiqSuite Workaround Status

The workaround should be implemented in the user application.

4.22 ERR082: Memory: Potential lockup when DC/GFX accesses MSPI/ Extended Memory while MSPI is DMAing to there

4.22.1 Description

There is a possibility of a lock up during concurrent MSPI and extended memory accesses by the Display Controller (DC) or the Graphics Module (GFX) when MSPI is doing DMA transfers to/from the same Extended Memory.

A scenario for this to happen is as follows:

1. An MSPI is doing DMA's to/from an Extended Memory space.
2. The GFX or DC does a read of the MSPI but cannot complete due to the active DMA operation in (1).
3. The GFX or DC does a read of the same Extended Memory space, but the GFX/DC cannot accept the read data in the FIFO because it must first complete the read in (2).

4.22.2 Affected Silicon Revisions

This silicon erratum applies to all silicon revisions of Apollo4 Plus SoC.

4.22.3 Application Impact

This issue can possibly affect user applications by causing a lock up.

4.22.4 Workarounds

The workaround for this deadlock situation is for the system configuration and/or program execution to not allow all three events to occur at once. It can be avoided by having a separate Extended Memory space for the GFX/DC's memory and MSPI memory.

4.22.5 Erratum Resolution Status

There are currently no plans to fix this erratum.

4.22.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK cannot prevent this issue from occurring - avoiding it must be done within the application design.

4.23 ERR085: IOS: Possible failure in FIFO mode in Wrap configuration

4.23.1 Description

When using the IOS's wrap configuration (CFG_WRAPPTR = WRAP), there are two conditions which may cause FIFO mode to fail because of a conflict with the special register space:

- FIFO_BASE is set to 0x10 (halfway into the LRAM) when FIFO reads are to be executed.
- FIFOPTR is set within the range 0x78 to 0x7F.

4.23.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.23.3 Application Impact

The impact of this issue on user applications is that FIFO read transfers will be corrupted because the LRAM address will not be incremented correctly. If FIFO_BASE is set to 0x10, the actual FIFO base address is 0x80. In Wrap mode the address pointer is set to the value of the host access and then mapped to the LRAM (i.e., FIFO space) by adding 8. Thus host accesses to the special space will cause a wraparound of the LRAM address which is not correct.

4.23.4 Workarounds

The workaround for this issue is to ensure that FIFO_BASE is not set to 0x10, and that the initial FIFOPTR is never set to an address within the special register space when operating in the Wrap configuration.

4.23.5 Erratum Resolution Status

There are no plans to fix this erratum.

4.23.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not perform any error checking to ensure that FIFO_BASE and FIFOPTR are configured properly in Wrap configuration.

4.24 ERR087: MCU_CTRL: POR failure due to VDDC/VDDF not rising to proper level

4.24.1 Description

A POR failure may occur during power-on. The POR design includes a level-shifter without isolation control and during power up, the uncontrolled level-shifter may cause loss of reset in the AOH domain. This in turn could incorrectly enable a strong pull-down path for the SIMOBUCK voltage. The MEMLDO will fail to power up VDDF because of the excessive loading when the SIMOBUCK voltage is pulled down enough.

4.24.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo4 Plus SoC.

4.24.3 Application Impact

The system may fail to power up properly when VDDC/VDDF does not reach specified minimum voltage during POR.

4.24.4 Workarounds

Recommended workaround is to connect a 2.2 uF capacitor between VDDF and VDD (VDD supply to VDDP/VDDH/VDDA).

4.24.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.24.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

4.25 ERR088: SDIO: DDR modes not supported

4.25.1 Description

The SDIO interface cannot support DDR mode as under certain circumstances the SDIO logic creates a transition on some of the data lines in DDR mode just as the clock is switching, which can cause hold time violations.

4.25.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of theApollo4 Plus SoC.

4.25.3 Application Impact

This issue affects user applications which were intended to use DDR mode to connect over SDIO. Data throughput is limited by the maximum specified clock frequency in SDR mode.

4.25.4 Workarounds

There is no workaround which enables the use of DDR mode. The SDIO interface should be used in SDR mode within the specified clock rate.

4.25.5 Erratum Resolution Status

There currently are no plans to fix this erratum on Apollo4 Plus SoC.

4.25.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

4.26 ERR090: ADC: No CNVCMP interrupt for first single scan

4.26.1 Description

When configured for single scan mode, the conversion complete (CNVCMP) bit is not set in ADC INTSTAT register, and the CNVCMP interrupt is not triggered, for the first scan. As well, there is no valid conversion result in the FIFO or FIFOPR register. Issue occurs at any settable clock source/rate, using any input channel, or using any slot.

4.26.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.26.3 Application Impact

This issue affects user applications which try to use the specified configuration.

4.26.4 Workarounds

Workarounds for this issue are as follows:

1. Use Single Scan Mode, set the slot config to average in 1 measurement (ADSELn=0), set ADCEN = 1 and trigger the ADC. The actual trigger source can be either software or external trigger. After the trigger, a FIFO pop must be performed in order to discard the errant sample. The workaround need not be reapplied as long as the ADC remains enabled (ADCEN=1).
2. Use Repeat Scan Mode with the internal repeating trigger timer (INTTRIGTMR_TIMEREN = EN). Using the internal repeating trigger timer automatically generates continuous conversion triggers. Even if the first trigger is missed, the following trigger will keep triggering the conversion until there is a valid result (and CNVCMP interrupt).

4.26.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.26.6 AmbiqSuite Workaround Status

The Ambiqsuite SDK 4.1.0 Apollo4b and Apollo4p HALs provide an implementation of the single scan workaround in the form of a call to am_hal_adc_power_control() with the ePowerState argument equal to AM_HAL_SYSCTRL_WAKE. This implementation powers up the ADC and performs the single scan workaround. Note also that in this implementation ADCEN is left enabled on exit from the function. This function may not be suitable for all applications, in which case the user will need to provide their own implementation.

4.27 ERR091: ADC: Loss of first scan data

4.27.1 Description

The first scan result sometimes gets lost. When ADC conversion completes (CNVCMP flag set), the conversion data is not sent to the FIFO.

4.27.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.27.3 Application Impact

This issue affects user applications which require a conversion result for the first scan.

4.27.4 Workarounds

The workaround for this issue is to do the following:

1. Use the slowest ADC clock setting: HFRC/2 (24MHz)
2. Use a TRKCYCx setting of 32 or higher.
3. Enable ADC auto calibration.

4.27.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.27.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK HAL does not provide a software workaround for this issue. The application should follow the above workaround.

4.28 ERR092: ADC/AUDADC: DMACPL interrupt sometimes fails

4.28.1 Description

The DMA complete (DMACPL) interrupt sometimes does not occur.

4.28.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.28.3 Application Impact

This issue has little effect on user applications provided the DMACPL interrupt is not used.

4.28.4 Workarounds

The workaround for this issue is to use the FIFOOVR1 interrupt instead of the DMACPL interrupt.

4.28.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.28.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides a software workaround for this issue and is implemented in the audadc_rtt_stream example in the SDK.

4.29 ERR094: BootROM: Does not accept source/destination address higher than 0x101D7FFF

4.29.1 Description

The current bootROM code is not updated for the SRAM size in Apollo4 Plus - it follows the Apollo4 definition of SRAM size based on CHIPREV, limiting it to 1888 KB. In the AmbiqSuite SDK, several API interfaces return error code 4 (pui32Src is invalid) when the pui32Src is located at an address higher than 0x101D7FFF. The bootROM helpers which take source/destination buffer in SRAM include:

- am_hal_mram_main_program
- am_hal_mram_info_program
- am_hal_mram_main_words_program
- am_hal_mram_info_read

4.29.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.29.3 Application Impact

This issue affects the execution of bootROM code in applications which currently use source/destination buffers in the higher SRAM address range.

4.29.4 Workarounds

The workaround for this issue is to avoid using MRAM HAL source and destination buffers in addresses higher than 0x101D7FFF.

4.29.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.29.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK. Applications must use buffers in the allowed space.

4.30 ERR096: DC: DPI-2 interface is not supported

4.30.1 Description

Use of the DPI-2 interface is not supported.

4.30.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.30.3 Application Impact

This issue limits display interface options in user applications to SPI, QSPI or DSI.

4.30.4 Workarounds

The workaround for this issue is to use the Display SPI, QSPI or DSI interface.

4.30.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.30.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides functionality and examples for the supported display interfaces.

4.31 ERR097: MSPI: Non-DQS SDR Octal not reliably operable at 48 MHz or 96 MHz

4.31.1 Description

In Apollo4 Plus, the MSPI does not operate reliably at 48 MHz and 96 MHz in octal, non-DQS SDR mode. Note that Octal at 96 Mhz in DQS SDR mode works as intended.

4.31.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.31.3 Application Impact

This issue requires that an application using Octal, non-SDR mode slows down the interface clock to lower than 48 MHz.

4.31.4 Workarounds

The workaround for this issue is to clock Octal, non-DQS SDR mode lower than 48 MHz or use DQS mode.

4.31.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.31.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK. Applications should use one of the workarounds mentioned above.

4.32 ERR098: STIMER: Constraints on writing to SCMPRn registers and handling Compare interrupts

4.32.1 Description

Compare interrupts are delayed by one STIMER clock. Additionally, on Apollo4 Plus it takes two STIMER clock cycles for the write to an SCMPRn register (where n is 0 to 7 representing one of the STIMER Compare registers) to get operated on. These timing issues put constraints on the minimum value of delta that can be applied to SCMPRn, which is two for Apollo4 and four for Apollo4 Plus.

In addition, back-to-back writes to SCMPRn may not work reliably (i.e., take the last value) on Apollo4 Plus unless the application ensures not to write within two STIMER clock cycles of the previous one. As well, after writing to SCMPRn, the application needs to wait for at least three STIMER clock cycles before reading it back for the new value to be reflected.

4.32.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.32.3 Application Impact

This issue may affect user applications by not generating an STIMER interrupt when it is expected, or with unreliable read/write of SCMPRn. Also, when updating the Compare interrupt time by writing to SCMPRn, it is possible that the application may still get a stale interrupt corresponding to the old value even after a new value is written to SCMPRn because of the latency with a write operation. This could happen if SCMPRn is written too close to the imminent interrupt.

4.32.4 Workarounds

The workaround for this potential issue is to ensure that the minimum delta for the next compare is specified correctly. Internal latencies must be accounted for by adjusting (reducing) the delta that is actually supplied in the SCMPRn register, keeping in mind that the programmed delta must be at least 1.

For Apollo4 Plus, this latency correction is 3. Therefore, the minimum valid delta setting is 4.

Also the application needs to handle back-to-back writes to SCMPRn and ensure not to write within two STIMER clock cycles of the previous one, and then wait for at least three STIMER clock cycles before reading it back for the new value to be reflected. Application also needs to handle the unlikely event of a stale Compare interrupt.

4.32.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.32.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK HAL will contain the workaround for these delays in a future release by internally adjusting the delta amount to be written to SCMPRn. Specifically, the HAL will account for the Compare interrupts being delayed by one STIMER clock by adjusting the delta value.

In addition, on Apollo4 Plus, the HAL will account for an extra 2 STIMER clock cycles of latency for the writes to SCMPRn by adjusting the delta value. The HAL will also handle the proper back-to-back writes to COMPARE and read-back, inserting waits if needed.

There is no workaround in the HAL for rare stale Compare interrupts, which could occur if SCMPRn is written too close to the imminent interrupt. Applications need to handle such a case.

4.33 ERR099: IOM: CQ does not pause via the BLE module

4.33.1 Description

The CQ is configured to pause on a BLE CQ Pause flag event, but CQPAUSE through BLE is not connected in the module design (IOM_CQPAUSEN_CQPEN = BLEXOREN has no effect).

4.33.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.33.3 Application Impact

This issue affects user applications by not supporting the pausing of the CQ via the BLE module.

4.33.4 Workarounds

There is no workaround for this issue.

4.33.5 Erratum Resolution Status

There currently are no plans to fix this erratum on Apollo4 Plus SoC, however it is planned to be fixed on future derivatives.

4.33.6 AmbiqSuite Workaround Status

This specific functionality is not supported in the AmbiqSuite SDK's HAL.

4.34 ERR101: IOM: Command write causes CQ operations to pause and never restart

4.34.1 Description

An invalid write of 0b'00 to the IOM Module's CMD_CMD field causes the command queue (CQ) to pause indefinitely.

Proper CQ operation states that if no command is started by the register write, the next doublet (address/data) will be fetched by the CQ. However, if the CQ is enabled with CQPAUSEEN = 0x0 (no pause events enabled), then the CQ starts processing instruction doublets. When the instruction to write 0x0 to the CMD_CMD field is executed, the CQ is paused (blocked waiting for a CMDCMP from the interface) and never restarts, even though writing 0x0 to this field should result in no command started.

4.34.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.34.3 Application Impact

This issue causes user applications to pause indefinitely when an invalid write to the CMD_CMD field occurs.

4.34.4 Workarounds

The workaround for this issue is to ensure that no write of 0x0 by the CQ to the CMD_CMD field occurs.

4.34.5 Erratum Resolution Status

There currently are no plans to fix this erratum on Apollo4 Plus SoC, however it is planned to be fixed on future derivatives.

4.34.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a software workaround for this issue or prevent the issue from occurring. It is the responsibility of the application to ensure that improper register writes by commands in the CQ do not occur.

4.35 ERR102: IOM: CQ does not pause immediately after triggering event

4.35.1 Description

The CQ is configured to pause on a GPIOXOREN event (CQPAUSE_CQPEN = GPIOXOREN), where the input GPIO irq_bit XORed with SWFLAG2 is '1' and the SWFLAG[2] path of the pause event is triggered.

A software-triggered CQ pause does not stop immediately upon write to the CQSETCLEAR register - one additional operation occurs after the register write. After hitting a pause event, the CQPAUSE bit is asserted and then de-asserted for 1 clock cycle which allows another CQ buffer entry to be executed.

4.35.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.35.3 Application Impact

This issue affects user applications by allowing execution of an additional CQ buffer operation after a write to the CQSETCLEAR register which may have an adverse affect on IOM CQ and application operation.

4.35.4 Workarounds

The workaround for this issue is to install an extraneous command in the CQ after a command that writes to the CQSETCLEAR field so that there are no adverse effects if the pause occurs after its execution.

4.35.5 Erratum Resolution Status

There currently are no plans to fix this erratum on Apollo4 Plus SoC, however it is planned to be fixed on future derivatives.

4.35.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a software workaround for this issue or prevent the issue from occurring. It is the responsibility of the application to ensure that execution of an additional CQ command after writing to the CQSETCLEAR field and before the pause occurs has no undesired effect.

4.36 ERR103: IOM: FIFO threshold interrupt incorrectly triggered

4.36.1 Description

In normal read operation, the FIFO threshhold interrupt (THR) and associated register bit (INTSTAT_THR) are asserted when the number of valid bytes in the read FIFO (FIFOPTR_FIFOnSIZ) equals or exceeds the value set in the read threshhold field (FIFOTHR_FIFORTHR), and similarly for write operation.

When the IOM is set up to do a read transaction (DMA or non-DMA), the FIFOWTHR trigger is gating the read logic.

The FIFOWTHR logic is not being qualified with a “write_event” and the FIFORTHR is not being qualified with a “read_event”. Therefore the FIFOWTHR is gating the read process.

4.36.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.36.3 Application Impact

This issue may affect user applications that do not evaluate actual FIFO pointers before initiating a FIFO read operation to determine how much data is ready to be read.

4.36.4 Workarounds

The workaround for this issue is for the application to always inspect the actual FIFO pointers before determining how much data is available to read. It is possible that a false threshold interrupt may be triggered which should be ignored.

4.36.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.36.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides all necessary memory and register access functions to enable the application to implement the stated workaround.

4.37 ERR104: IOM: Data corrupted on I2C when OFFSETCNT=0 and I2CLSB=1

4.37.1 Description

The first data byte sent over I2C is sent most significant bit (MSB) first when MI2CCFG_I2CLSB = LSBFIRST and CMD_OFFSETCNT = 0. If OFFSETCNT = 0, all data shifted out should be LSB first but the first byte of the transfer is sent MSB first and the remaining 3 bytes are LSB first.

4.37.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.37.3 Application Impact

This issue may affect user applications and proper I2C data interpretation if LSB first configuration is used.

4.37.4 Workarounds

A workaround for this issue is to avoid it by not configuring the I2C transfers as LSB first, if possible.

4.37.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.37.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a software workaround for this issue or prevent the issue from occurring as it does not limit I2C use or configuration. It is the responsibility of the application to avoid or handle this issue.

4.38 ERR105: GPU: System hang when GPU fetches data from internal SRAM

4.38.1 Description

When the GPU fetches data from internal SRAM, a system hang occurs. Graphics output stops during execution of the nema_get_cl_status function due to the statement if (last_cl_id >= cl_id) always returning false. The CL_ID is always 1 higher than LAST_CL_ID.

Output of the same buffer to the screen still occurs when forcing it past this point in the user graphics state buffer. However, it never moves past this issue of CL_ID always being 1 higher than LAST_CL_ID and a hang occurs at this point in the graphics.

Also, running full speed (no delays) causes the GPU to hang and graphics output to stop.

4.38.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.38.3 Application Impact

This issue adversely affects user applications if enabling the GPU to try to access data in internal SRAM.

4.38.4 Workarounds

The workaround for this issue requires two steps:

1. Modify am_memory_map.h to specify new locations of variables from SRAM to SSRAM.
2. Add delays to prevent CL_ID from getting stuck.

4.38.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.38.6 AmbiqSuite Workaround Status

A non-blocking graphics example with proper memory allocation/use is provided in nemagfx_balls_bench_nb.

4.39 ERR106: AUDADC: MCU hangs when attempting to configure with disabled XTALHS clock

4.39.1 Description

The following sequence will result in MCU hang:

1. The high-speed crystal (XTALHS) is enabled.
2. The AUDADC module is enabled and configured to be clocked with the XTALHS_24MHz clock option by setting the AUDADC_CFG_CLKSEL field.
3. The MCU undergoes a SWPOR reset *without* setting the AUDADC clock source to OFF, HFRC_48MHz or HFRC2_48MHz.
4. The AUDADC is attempted to be enabled without first enabling the XTALHS clock source.

Since the AUDADC is attempted to be re-enabled using its prior configuration of using the XTALHS_24MHz clock option without first enabling the high-speed crystal again after the reset, the MCU hangs.

The AUDADC clock selection persists through a Software Power-On Reset instead of having its clock revert to the default value of OFF (no clock selected). A full power-on reset (POR) does set the clock initially to OFF as expected and no hang is experienced when enabling the AUDADC.

4.39.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.39.3 Application Impact

This issue may affect user applications that clock the AUDADC module with the high-speed crystal.

4.39.4 Workarounds

The workarounds for this issue are as follows:

1. To fully reset the AUDADC to its default settings, perform a full POR which sets the AUDADC clock to OFF (no clock selected).
2. When powering down the AUDADC, set the module's clock to OFF if the.clock selection was the XTALHS. Then there will be no problem when the AUDADC is next initialize and powered on. The module's clock source may then be reselected upon re-enabling the AUDADC, making sure that the XTALHS is enabled if it is intended to be the clock source.

4.39.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.39.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides the procedure to properly enable clocks and configure the AUDADC with the desired clock source.

4.40 ERR109: MSPI: Dedicated DQS falling edge delay settings do not work in DEV0DDR register

4.40.1 Description

For any of the three MSPI instances, setting the DEV0DDR's RXDQSDELAYNEGEN0 field and the RXDQSDELAYNEG0 field with a delay offset for the falling edge of the DM0/DQS0 signal does not work and has been deprecated. If a value is programmed in the RXDQSDELAY0 register, then this value sets the delay offset for both rising and falling edges of DM0/DQS0. Additionally, the setting of the ENABLEFINEDELAY0 bit has no effect when enabling the DM0/DQS0 delay. Enabling the programmed delay for DM0/DQS0 is done simply by writing a delay value to the RXDQSDELAY0 field.

For MSPI0 (the only MSPI instance that supports hex mode), programming the RXDQSDELAYNEGHI0 field with a delay offset for the falling edge of the DM1/DQS1 signal does not work and has been deprecated. If the RXDQSDELAYHIEN0 bit is set and the RXDQSDELAYHI0 field has been programmed with a delay offset for the rising edge of the DM1/DQS1 signal, the same delay will apply to the falling edge.

In summary, the ENABLEFINEDELAY0, RXDQSDELAYNEGEN0, RXDQSDELAYNEG0, and RXDQSDELAYNEGHI0 fields have no effect and should remain at their default values (0).

4.40.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.40.3 Application Impact

This issue should not have an adverse effect on user applications, as adequate timing can be achieved without the use of separate falling edge delay offsets.

4.40.4 Workarounds

The workaround for this issue is to fine-tune the delay of RXDQSDELAY0 and RXDQSDELAYHI0 for best timing of both rising and falling edges of their corresponding DQS signals.

4.40.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.40.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides the capability to implement the workaround mentioned above to set the timing of the DQS signal(s) adequately.

4.41 ERR110: DAXI: Out-of-order SSRAM read and write returns incorrect read value

4.41.1 Description

Issue Summary:

An out-of-order read of Shared SRAM (SSRAM) may occur when an SSRAM read reaches the SRAM-AXI Read Address FIFO while there is an earlier write to that same address in the SSRAM-AXI Write Address FIFO which has not yet completed.

The effect of this mis-ordering of an SSRAM address read and write is that the read returns the prior value of the SSRAM address before the write has taken place.

Issue Root Cause:

The root cause is that the DAXI allows a read transaction to go out to the same SSRAM address for which there is an in-flight write transaction.

Condition under which issue may occur:

This issue occurs only under the rare condition where write data to SSRAM is delayed from reaching the SSRAM in the DAXI-AXI FIFO, behind slower MSPI XIP memory-mapped write data. This primarily occurs when MSPI accepts a write address, but its write data buffers are full.

The SSRAM has separate read and write address FIFOs which make the SSRAM susceptible to mis-ordering of the read and write operations. However, writes to SSRAM complete as fast as the write addresses can reach the SSRAM FIFO, unless the data for the write is delayed. This issue therefore will not occur if CPU is only writing to SSRAM and/or ESRAM.

Note that the Extended SRAM (ESRAM) and MSPI have combined read/write address FIFOs, so they are not susceptible to this issue.

4.41.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.41.3 Application Impact

This issue affects user applications by returning incorrect SSRAM data under certain conditions.

4.41.4 Workarounds

The condition under which this issue occurs is rare and can be further minimized, but not eliminated, by locating stacks and data accessed within ISRs in TCM and ESRAM (not SSRAM).

The issue may be completely avoided by not mixing CPU XIP memory-mapped writes to MSPI with CPU writes to SSRAM.

A flush of DAXI must be performed when switching between allowing CPU XIP memory-mapped writes to MSPI and allowing CPU writes to SSRAM. A hardware DAXI flush can be used. For acceptable performance, most or all stacks should be located in Tightly Coupled Memory (TCM) or ESRAM.

One method of separating CPU writes to MSPI from CPU writes to SSRAM is to use the MPU and memory region definitions to enforce that the CPU does not write to one range while writing to another range is allowed.

4.41.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.41.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a prevention for this issue. It is the responsibility of the application to avoid or handle this issue by the use of the described workaround.

4.42 ERR111: MSPI: Delayed MSPI write b-response may cause MSPI state machine deadlock

4.42.1 Description

Background:

For every write across the AXI bus, a b-response handshake takes place to confirm to the master that the write has completed successfully. When a write has completed transfer to the target device's FIFOs (address and data), the device puts a b-response on the AXI bus. The master reads the b-response and provides a return b-response to the target. Once the b-response handshake is complete, the target accepts the next transaction (if any).

Issue Summary:

1. Acceptance of an MSPI b-response by DAXI is delayed when the DAXI b-response FIFO becomes full. XIP memory-mapped writes to MSPI complete too slowly for the DAXI b-response FIFO to fill on their own, so filling the DAXI b-response FIFO can only occur if there are also some concurrent CPU writes to SSRAM or ESRAM.
2. While MSPI b-response is delayed and another master performs an XIP memory-mapped read from or write to MSPI, the MSPI b-response ID changes to this new master even though the b-response to DAXI has not yet taken place. The b-responses complete in order, so any writes to SSRAM or ESRAM are blocked from completing and will start to fill up the SSRAM and ESRAM Address FIFOs.

The result is that the AXI bus deadlock results in a CPU hang, which is typically detected by the Watchdog Timer in most applications.

Issue root cause:

The MSPI Interface does not enforce the correct b-response ID to be maintained on the AXI bus until the master accepts and responds to the b-response.

Condition under which issue may occur:

The condition under which this issue occurs is rare. Normally, the master accepts and responds to the b-response within a clock cycle of the MSPI making the b-response available on the AXI bus. However, if the master's b-response FIFO is full, then there is a delay in accepting and responding to the b-response.

The DAXI master requires 6 cycles to clear its 2-deep b-response FIFO, so its FIFO may be filled if there are more than 2 writes completed every 6 cycles. The MSPI is not capable of completing writes that quickly, so XIP memory-mapped writes to MSPI alone work correctly. CPU writes to SSRAM or ESRAM may be completed in < 3 cycles, so a burst of writes to SSRAM or ESRAM may fill the b-response FIFO. A write to MSPI after a burst of writes to SSRAM/ESRAM may then have its b-response delayed.

In such a case, while the MSPI b-response is delayed, if another master performs a read from or write to MSPI, the MSPI b-response ID changes to this new master even though the target's b-response to the DAXI has not yet completed. This prevents the MSPI b-response from reaching the DAXI.

Since the b-responses are enforced by the AXI bus to complete in order, any subsequent writes to SSRAM or ESRAM are blocked from completing and will start to fill up the SSRAM and ESRAM Address FIFOs.

If the issue was triggered by another master writing to MSPI, the MSPI state machine becomes deadlocked and cannot be recovered due to the b-response to the DAXI being delayed. The Issue becomes unrecoverable because the DAXI-AXI FIFO becomes blocked by writes to SSRAM/ESRAM, as this prevents any DAXI reads or writes from reaching MSPI to clear the issue.

If the issue was triggered by another master reading from MSPI, then the b-response ID issue may be resolved if a DAXI read from or write to MSPI reaches MSPI.

4.42.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.42.3 Application Impact

The occurrence of this issue may cause a CPU hang. However, the hang is recoverable with the use of the Watchdog Timer.

4.42.4 Workarounds

There are two workaround options available:

Option 1:

Prevent masters other than CPU from accessing MSPI memory (read or write) while the CPU is writing to MSPI.

DAXI Flush must be performed before allowing other masters to access MSPI to ensure any in-flight writes to MSPI complete before other masters access MSPI.

Option 2:

Avoid mixing CPU writes to MSPI with CPU writes to SSRAM and ESRAM.

A flush of DAXI must be performed when switching between allowing CPU writes to MSPI and allowing CPU writes to SSRAM/ESRAM. A hardware DAXI flush can be used. For acceptable performance, most or all stacks should be located in Tightly Coupled Memory (TCM).

One method of separating CPU writes to MSPI from CPU writes to SSRAM/ESRAM is to use the MPU and memory allocation adjustments to enforce that the CPU does not write to one range while writing to another range is allowed.

Note that Option 2 is a superset of the workaround for ERR110, and will work as a workaround for both ERR110 and ERR111, if the additional restrictions on ESRAM as described here can be accepted.

4.42.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.42.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a prevention for this issue. It is the responsibility of the application to avoid or handle this issue by the use of one of the described workarounds.

4.43 ERR112: ADC: Dummy trigger causes immediate (invalid) interrupt

4.43.1 Description

A CNVCMP interrupt asserts immediately after enabling the NVIC of the ADC even though the CNVCMP interrupt status flag remains cleared. This dummy trigger occurs without enabling the ADC CNVCMP interrupt and clearing the interrupt before enabling the ADC EN bit.

4.43.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Plus SoC.

4.43.3 Application Impact

This issue affects user applications if the dummy triggered interrupt is processed as a valid interrupt.

4.43.4 Workarounds

The workaround for this issue is to disregard the interrupt if no ISR status bits are set when the ISR is asserted.

4.43.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.43.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK HAL does not provide a software workaround for this issue. The application should follow the above workaround.

4.44 ERR113: ADC: Occasional corrupt conversion results at 48 MHz

4.44.1 Description

Conversion data is sometimes corrupted when operating the ADC at 48 MHz clock.

4.44.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of the Apollo4 Plus SoC.

4.44.3 Application Impact

This issue affects user applications by occasionally yielding incorrect conversion result.

4.44.4 Workarounds

The workaround for this issue is to use an ADC clock setting of 24 MHz (ADC_CFG_CLKSEL = HFRC_24MHz). In addition, setting *additional* input signal sampling/tracking time of at least 32 ADC clock cycles (SLnCFG_TRKCYCn = 32), for a total of 37 sampling/tracking cycles, should be used.

Note: Because of this erratum, 48 MHz ADC clock is no longer supported on the Apollo4 SoC. It continues to not be supported on the Apollo4 Plus SoC.

4.44.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.44.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK HAL does not provide a software workaround for this issue. The application should follow the above workaround.

5. Ordering Information

Table 4: SoC Ordering Information

Device Name	Orderable Part Number	MRAM	RAM	Package (mm)	Packing	Temperature Range
Apollo4 Plus SoC	AMAP42KP-KBR	2 MB	2.75 MB	5.0 x 5.0 146-pin BGA	Tape and Reel	–20 to 60°C
Apollo4 Blue Plus SoC	AMA4B2KP-KBR	2 MB	2.75 MB	4.7 x 4.7 131-pin BGA	Tape and Reel	–20 to 60°C
Apollo4 Blue Plus SoC	AMA4B2KP-KXR	2 MB	2.75 MB	4.7 x 4.7 131-pin BGA	Tape and Reel	–20 to 60°C

Ambiq Micro, Inc.
6500 River Place Boulevard, Building 7,
Suite 200, Austin, TX 78730-1156
www.ambiq.com/
sales@ambiqmicro.com
https://support.ambiqmicro.com
+1 (512) 879-2850

SE-A4P-3p0
Version 3.0
October 2022