ERRATA LIST

Apollo4 SoC, Apollo4 Blue SoC

Ultra-low Power Apollo SoC Family

Doc. ID: SE-A4-13p0

Doc. Revision: 13.0, June 2024

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

Introduction 5
Document Revision History 5
Errata Summary List 8
Detailed Silicon Errata 13
ERR002: ADC/AUDADC: DMA ISR cannot be properly triggered 14
ERR003: CACHE: Some modes are unsupported 15
ERR008: GPIO: Dual-edge interrupts are not vectoring 16
ERR014: IOS: SLINT does not occur as expected after IOINTSET command from host 17
ERR036: TIMER: First bit is lost after first trigger in SINGLEPATTERN mode 18
ERR038: IOM: CQ fails to complete DMA read transfers 19
ERR039: IOS: MISO line is not tri-stated when CE driven high 20
ERR040: IOS: FIFO read gets stuck/stalled 21
ERR041: USB: Induced D+ output pulse may cause unintended disconnect 22
ERR042: MSPI: Violates DMATIMELIMIT0 value if transaction is paused 23
ERR043: DMA: Incorrect reads/writes near memory boundaries 24
ERR046: GPIO: FIEN/FOEN not operational on GPIO0 25
ERR049: GPU: GPU hangs when using multiple textures in MRAM 26
ERR050: MCU_CTRL: High Performance mode entry failure 27
ERR052: MRAM: MRAM access may be corrupted during Crypto power up/down or OTP access 28
ERR053: SDIO: Incorrect data is returned on SDIO buffer reads 30
ERR056: USB: High leakage current in USB PHY 31
ERR059: TIMER: Functional limitations on all or specific timers 32
ERR063: MSPI: Interface may shut down too early upon RXF condition in non-DQS mode 35
ERR064: IOM: I2C power save/restore failure 36
ERR065: MSPI: CM4 hard fault not triggered when it should be 37
ERR066: IOS: Fails in FIFO mode at specific range of clock frequencies 38
ERR067: MSPI: RXCAP not operable when CLKDIV=1 39
ERR070: I2S: I2S0's ASRC Rx is non-operational 40
ERR071: INFO: Reads of INFO space fail 41
ERR073: GPU: Blit overruns destination texture 42
ERR074: DSI: Image displayed outside the set region 43
ERR075: I2S: Clocks incorrectly gated while in deep sleep 44
ERR076: MCU_CTRL: Failure to switch from High Performance to Low Power Modes 45
ERR077: DSI: Cannot free DBI when reading DSI register 46
ERR078: MSPI: Potential race condition when using RXNEG and RXDQSDELAY concurrently 47
ERR079: INFO0: SIMO Buck cannot be enabled via INFO0 setting 48
ERR080: MSPI: Command Queue may disable DMAEN while data is still pending in internal buffer 49
ERR081: Core: Device hangs when an asynchronous interrupt occurs just after entering deepsleep 50
ERR082: Memory: Potential lockup when DC/GFX accesses MSPI/ Extended Memory while MSPI is DMAing to there 51
ERR083: CLKGEN: HFRC adjustment sometimes does not work 52
ERR084: TIMER: STIMER capture/compare event cannot be used as a Timer trigger source 53
ERR085: IOS: Possible failure in FIFO mode in wrap configuration 54
ERR086: TIMER: Trigger by another timer output doesn't work as intended 55
ERR087: MCU_CTRL: POR failure due to VDDC/VDDF not rising to proper level 56
ERR090: ADC: No CNVCMP interrupt for first single scan 57
ERR091: ADC: Loss of first scan data 58
ERR096: DC: DPI-2 interface is not supported 59
ERR098: STIMER: Constraints on writing to SCMPRn registers and handling Compare interrupts 60
ERR099: IOM: CQ does not pause via the BLE module 62
ERR100: TIMER: High Deep Sleep current when TMR6/TMR9 enabled 63
ERR101: IOM: Command write causes CQ operations to pause and never restart 64

ERR102: IOM: CQ does not pause immediately after triggering event 65
ERR103: IOM: FIFO threshold interrupt incorrectly triggered 66
ERR104: IOM: Data corrupted on I2C when OFFSETCNT=0 and I2CLSB=1 67
ERR106: AUDADC: MCU hangs when attempting to configure with disabled XTALHS clock 68
ERR107: RTC: Clock domain crossing issue causes APB bus hang 70
ERR108: MSPI: Timing issue using CLKOND4 71
ERR110: DAXI: Out-of-order SSRAM read and write returns incorrect read value 72
ERR111: MSPI: Delayed MSPI write b-response may cause MSPI state machine deadlock 74
ERR112: ADC: Dummy trigger causes immediate (invalid) interrupt 76
ERR113: ADC: Occasional corrupt conversion results at 48 MHz 77
ERR114: BLE: Performance not guaranteed during concurrent USB operation 78
ERR115: GPIO: Possible glitch on GPIO outputs upon initial power up 80
ERR118: DC: Cannot read data from display panel in SPI mode 81
ERR119: ADC: Incorrect sample rate when using Internal ADC Timer as repeat clock source 82
ERR121: CLKGEN: XTAL32K is activated when XTAL_HS is selected as module clock 83
ERR123: TIMER: Timer module reset required between HW trigger-initiated timer operations 84
ERR124: MSPI: Mixed Mode 1-1-4 does not work as expected 85
ERR125: BLE: Corrupted non-volatile memory prevents boot-up of the BLE controller 86
ERR126: CLKGEN: HFADJ enabled with no HFRC-clocked modules powered causes incorrect HFRC clock frequency 87
ERR127: BLE: Image corruption at boot or reset 88
ERR128: MSPI: D3:D1 lines are pulled low instead of staying in high impedance mode 89
ERR129: RTC: CB field value is unpredictable when year = 99 and CEB = 1. 90
ERR130: CLOCKGEN: HF2ADJ-introduced jitter may cause incorrect HFRC2 adjustment 91
Ordering Information 93

Silicon Errata for the Apollo4 SoC and Apollo4 Blue SoC

1. Introduction

This document is a detailed compilation of known device errata for the general availability revision of the Apollo4 SoC and Apollo4 Blue 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
7.0	Sep 2021	**Initial General Release Version** Updated errata: • ERR036: Updated Erratum Resolution Status. • ERR066: Description, Workaround and AmbiqSuite Workaround Status fields updated. Added errata: • ERR079 - ERR086
8.0	Jan 2022	Updated errata: • ERR002: Updated to apply to AUDADC as well as ADC. • ERR014: Updated Workaround. • ERR063: Updated the AmbiqSuite Workaround Status. • ERR071: Updated the AmbiqSuite Workaround Status. Removed errata: • ERR024 (Feature Removed) • ERR031 (Feature Removed) Added errata: • ERR087, ERR090, ERR091, ERR092, ERR096, ERR098
9.0	June 2022	Updated errata: • ERR065: Updated title and application impact • ERR091: Updated with expanded workaround. Added errata: • ERR099 - ERR104, ERR106 - ERR108
10.0	Oct 2022	Updated errata: • ERR091: Removed note in Workaround regarding ADC CLKSEL HFRC/2 not offered on Apollo4. HFRC/2 should be used for the ADC CLKSEL. Added errata: • ERR110 - ERR113

Table 1: Document Revision History

Rev No.	Date	Description
11.0	Jul 2023	Updated errata: • ERR056: Updated pull-down requirement in workaround to 2 Mohm. Updated the AmbiqSuite Workaround Status to note that power sequencing has been updated in dsc_apollo4.c.in SDK version 4.4.0. • ERR071: AmbiqSuite Workaround Status updated. • ERR074: AmbiqSuite Workaround Status updated. Added errata: • ERR114, ERR115, ERR118, ERR121, ERR123, ERR124
12.0	Feb 2024	Updated errata: • ERR002: Updated Erratum Resolution Status • ERR003: Updated Description and Erratum Resolution Status • ERR014: Updated Erratum Resolution Status • ERR036: Updated Erratum Resolution Status • ERR040: Updated Erratum Resolution Status • ERR041: Updated Workarounds field • ERR042: Updated Erratum Resolution Status • ERR050: Updated Erratum Resolution Status • ERR052: Updated Erratum Resolution Status • ERR053: Updated Erratum Resolution Status • ERR059: Updated Erratum Resolution Status • ERR063: Updated Erratum Resolution Status • ERR064: Updated AmbiqSuite Workaround Status • ERR065: Updated Erratum Resolution Status • ERR066: Updated Description, Resolution and Erratum Resolution Status fields. • ERR067: Updated Erratum Resolution Status • ERR070: Updated Description and Erratum Resolution Status • ERR071: Updated Erratum Resolution Status • ERR073: Updated Erratum Resolution Status • ERR074: Updated Erratum Resolution Status • ERR075: Updated Erratum Resolution Status • ERR076: Updated Erratum Resolution Status • ERR078: Updated Erratum Resolution Status • ERR080: Updated Erratum Resolution Status • ERR081: Updated Erratum Resolution Status • ERR083: Updated Erratum Resolution Status • ERR084: Updated Erratum Resolution Status • ERR086: Updated Erratum Resolution Status • ERR091: Updated Workarounds section. • ERR098: Clarified Description, Workarounds and AmbiqSuite Workaround Status fields. • ERR099: Updated Erratum Resolution Status • ERR100: Updated Erratum Resolution Status • ERR101: Updated Erratum Resolution Status • ERR107: Updated Erratum Resolution Status Added errata: • ERR125

Table 1: Document Revision History

Rev No.	Date	Description
13.0	June 2024	Removed errata: • ERR092 (Combined with ERR002) Updated errata: • ERR002: Updated to also apply to AUDADC. • ERR090: Updated Workaround. • ERR091: Updated Workaround. • ERR098: Updated Description and AmbiqSuite Workaround Status. • ERR113: Updated Workaround and AmbiqSuite Workaround Status. Added errata: • ERR119, ERR126 - ERR130

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 versions implies that those fixes are on the latest Apollo4 SoC and Apollo4 Blue SoC revision as well unless otherwise stated.

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
“ERR002: ADC/AUDADC: DMA ISR cannot be properly triggered” on page 14	All existing	No Fix planned	Software workaround
“ERR003: CACHE: Some modes are unsupported” on page 15	All existing	Partially fixed; no further fix planned.	No workaround
“ERR008: GPIO: Dual-edge interrupts are not vectoring” on page 16	All existing	No fix planned	Software and/or hardware workaround
“ERR014: IOS: SLINT does not occur as expected after IOINTSET command from host” on page 17	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR036: TIMER: First bit is lost after first trigger in SINGLEPATTERN mode” on page 18	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR038: IOM: CQ fails to complete DMA read transfers” on page 19	All existing	No fix planned	Software workaround
“ERR039: IOS: MISO line is not tri-stated when CE driven high” on page 20	All existing	No fix planned	Software/hardware workaround
“ERR040: IOS: FIFO read gets stuck/stalled” on page 21	All existing	Partially fixed; no further fix planned.	No workaround
“ERR041: USB: Induced D+ output pulse may cause unintended disconnect” on page 22	All existing	No fix planned	Software/hardware workaround
“ERR042: MSPI: Violates DMATIMELIMIT0 value if transaction is paused” on page 23	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR043: DMA: Incorrect reads/writes near memory boundaries” on page 24	All existing	No fix planned	Software workaround
“ERR046: GPIO: FIEN/FOEN not operational on GPIO0” on page 25	All existing	No fix planned	Software/hardware workaround
“ERR049: GPU: GPU hangs when using multiple textures in MRAM” on page 26	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR050: MCU_CTRL: High Performance mode entry failure” on page 27	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
“ERR052: MRAM: MRAM access may be corrupted during Crypto power up/down or OTP access” on page 28	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR053: SDIO: Incorrect data is returned on SDIO buffer reads” on page 30	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR056: USB: High leakage current in USB PHY” on page 31	All existing	No fix planned	Hardware workaround
“ERR059: TIMER: Functional limitations on all or specific timers” on page 32	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR063: MSPI: Interface may shut down too early upon RXF condition in non-DQS mode” on page 35	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR064: IOM: I2C power save/restore failure” on page 36	All existing	No fix planned	Software workaround
“ERR065: MSPI: CM4 hard fault not triggered when it should be” on page 37	All existing	No fix planned	Software workaround
“ERR066: IOS: Fails in FIFO mode at specific range of clock frequencies” on page 38	All existing	No fix planned	No workaround
“ERR067: MSPI: RXCAP not operable when CLK-DIV=1” on page 39	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR070: I2S: I2S0’s ASRC Rx is non-operational” on page 40	All existing	No fix planned for Apollo4. Fixed on Plus.	No workaround
“ERR071: INFO: Reads of INFO space fail” on page 41	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR073: GPU: Blit overruns destination texture” on page 42	All existing	No fix planned	Software workaround
“ERR074: DSI: Image displayed outside the set region” on page 43	All existing	No fix planned for Apollo4. Fixed on Plus.	Software workaround
“ERR075: I2S: Clocks incorrectly gated while in deep sleep” on page 44	All existing	No fix planned	Limited software work-around
“ERR076: MCU_CTRL: Failure to switch from High Performance to Low Power Modes” on page 45	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR077: DSI: Cannot free DBI when reading DSI register” on page 46	All existing	No fix planned	Software workaround
“ERR078: MSPI: Potential race condition when using RXNEG and RXDQSDELAY concurrently” on page 47	All existing	No fix planned	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
“ERR079: INFO0: SIMO Buck cannot be enabled via INFO0 setting” on page 48	All existing	No fix planned	Software workaround
“ERR080: MSPI: Command Queue may disable DMAEN while data is still pending in internal buffer” on page 49	All existing	No fix planned	Software workaround
“ERR081: Core: Device hangs when an asynchronous interrupt occurs just after entering deepsleep” on page 50	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR082: Memory: Potential lockup when DC/GFX accesses MSPI/ Extended Memory while MSPI is DMAing to there” on page 51	All existing	No fix planned	System configuration
“ERR083: CLKGEN: HFRC adjustment sometimes does not work” on page 52	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR084: TIMER: STIMER capture/compare event cannot be used as a Timer trigger source” on page 53	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR085: IOS: Possible failure in FIFO mode in wrap configuration” on page 54	All existing	No fix planned	Software workaround
“ERR086: TIMER: Trigger by another timer output doesn’t work as intended” on page 55	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR087: MCU_CTRL: POR failure due to VDDC/ VDDF not rising to proper level” on page 56	All existing	No fix planned	Hardware workaround
“ERR090: ADC: No CNVCMP interrupt for first single scan” on page 57	All existing	No fix planned	Software workaround
“ERR091: ADC: Loss of first scan data” on page 58	All existing	No fix planned	Software workaround
“ERR096: DC: DPI-2 interface is not supported” on page 59	All existing	No fix planned	Software workaround
“ERR098: STIMER: Constraints on writing to SCMPRn registers and handling Compare interrupts” on page 60	All existing	No fix planned	Software workaround
“ERR099: IOM: CQ does not pause via the BLE module” on page 62	All existing	No fix planned	No workaround
“ERR100: TIMER: High Deep Sleep current when TMR6/TMR9 enabled” on page 63	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Hardware workaround
“ERR101: IOM: Command write causes CQ operations to pause and never restart” on page 64	All existing	No fix planned	Software workaround
“ERR102: IOM: CQ does not pause immediately after triggering event” on page 65	All existing	No fix planned	Software workaround
“ERR103: IOM: FIFO threshold interrupt incorrectly triggered” on page 66	All existing	No fix planned	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
“ERR104: IOM: Data corrupted on I2C when OFFSETCNT=0 and I2CLSB=1” on page 67	All existing	No fix planned	Software workaround
“ERR106: AUDADC: MCU hangs when attempting to configure with disabled XTALHS clock” on page 68	All existing	No fix planned	Software workaround
“ERR107: RTC: Clock domain crossing issue causes APB bus hang” on page 70	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	Software workaround
“ERR108: MSPI: Timing issue using CLKOND4” on page 71	All existing	No fix planned	Hardware workaround
“ERR110: DAXI: Out-of-order SSRAM read and write returns incorrect read value” on page 72	All existing	No fix planned	Software workaround
“ERR111: MSPI: Delayed MSPI write b-response may cause MSPI state machine deadlock” on page 74	All existing	No fix planned	Software workaround
“ERR112: ADC: Dummy trigger causes immediate (invalid) interrupt” on page 76	All existing	No fix planned	Software workaround
“ERR113: ADC: Occasional corrupt conversion results at 48 MHz” on page 77	All existing	No fix planned	Software workaround
“ERR114: BLE: Performance not guaranteed during concurrent USB operation” on page 78	All existing	No fix planned	No workaround
“ERR115: GPIO: Possible glitch on GPIO outputs upon initial power up” on page 80	All existing	No fix planned for Apollo4. Fixed on Plus and Lite.	No workaround
“ERR118: DC: Cannot read data from display panel in SPI mode” on page 81	All existing	Fix planned on future SoC family	No workaround
“ERR119: ADC: Incorrect sample rate when using Internal ADC Timer as repeat clock source” on page 82	All existing	No fix planned	Software workaround
“ERR121: CLKGEN: XTAL32K is activated when XTAL_HS is selected as module clock” on page 83	All existing	No fix planned	No workaround
“ERR123: TIMER: Timer module reset required between HW trigger-initiated timer operations” on page 84	All existing	No fix planned for Apollo4 or Plus. Fixed on Lite.	Software workaround
“ERR124: MSPI: Mixed Mode 1-1-4 does not work as expected” on page 85	All existing	No fix planned	Software workaround
“ERR125: BLE: Corrupted non-volatile memory prevents boot-up of the BLE controller” on page 86	Blue versions only	No fix planned	Hardware workaround
“ERR126: CLKGEN: HFADJ enabled with no HFRC-clocked modules powered causes incorrect HFRC clock frequency” on page 87	All existing	No fix planned	Software workaround
“ERR127: BLE: Image corruption at boot or reset” on page 88	All existing	No fix planned	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
“ERR128: MSPI: D3:D1 lines are pulled low instead of staying in high impedance mode” on page 89	All existing	No fix planned	Hardware workaround
“ERR129: RTC: CB field value is unpredictable when year = 99 and CEB = 1.” on page 90	All existing	No fix planned	Software workaround
“ERR130: CLOCKGEN: HF2ADJ-introduced jitter may cause incorrect HFRC2 adjustment” on page 91	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/AUDADC: DMA ISR cannot be properly triggered

4.1.1 Description

The ADC/AUDADC DMA ISR cannot be properly triggered and results in missing interrupt conditions for the DMA Transfer Complete (DCMP) and DMA ERROR (DERR) interrupts.

4.1.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.1.3 Application Impact

The 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 issue is to use the FIFO 75% full interrupt (FIFOOVR1), triggered when the INTSTAT_FIFOOVR1 status bit is set, instead of using the INTSTAT_CNVCMP interrupt. Inside the ISR, read the DMA_STAT_DMACPL bit until it is set indicating the DMA transfer is complete.

4.1.5 Erratum Resolution Status

There are no plans at this time to fix this erratum on Apollo4 family SoCs.

4.1.6 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

4.2 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 5 are supported:

- W1_128B_512E
- W2_128B_512E
- W1_128B_1024E
- W2_128B_2048E
- 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 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

There are no plans at this time to fix this erratum on Apollo4 family SoCs.

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 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 ERR014: IOS: SLINT does not occur as expected after IOINTSET command from host

4.4.1 Description

The SLINT is not seen after the IOINTSET command from the host. There are actually two issues here:

- IOS slave does not generate an interrupt to the host when initiated by the CPU (by writing to IOINTCTL_IONTSET).
- IOS slave does not generate an interrupt to the host when the host writes to the IOINT registers over SPI/I2C. A CPU interrupt is expected to be generated whenever the host writes any IOINT register.

4.4.2 Affected Silicon Revisions

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

4.4.3 Application Impact

The host will not get an interrupt when the slave writes to the IOINTCTL_IONTSET field, or when the host writes to any IOINTCTL register (over SPI/I2C). There is no negative impact to user applications if a GPIO is manually triggered as recommended in the below workaround.

4.4.4 Workarounds

A workaround for this limitation is to control an available GPIO by software to signal an interrupt.

4.4.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.4.6 AmbiqSuite Workaround Status

A software workaround is to use an available pin as a standard GPIO (instead of SLINT) and manipulate this inside the application code. This is demonstrated in the ios_fifo example.

4.5 ERR036: TIMER: First bit is lost after first trigger in SINGLEPATTERN mode

4.5.1 Description

In SINGLEPATTERN mode, bits stored in CMP0 and CMP1 registers are shifted to form the pattern of the output on OUT0 and OUT1. A trigger can be selected to enable the same pattern to repeat on the outputs when the trigger occurs. However, after the initial trigger which produces the correct pattern output, the output occurs on subsequent triggers without the first bit (bit 0 of CMP0, or CMP0 and CMP1 depending on the setting of the CTRLn_TMRnLMT field).

The issue occurs on all timer channels in SINGLEPATTERN mode, regardless of the setting of the CTRLn_TMRnLMT field. Also, it occurs for any trigger input or trigger edge selected.

4.5.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.5.3 Application Impact

For applications using this timer mode, the output pattern will not be correct or as expected after the initial triggered output. On subsequent triggered output patterns, the first bit of the pattern will not be output.

4.5.4 Workarounds

If using repeat triggers, the workaround for this output behavior is to set up the desired output pattern to start at bit 1 of the CMP0 and/or CMP1 register instead of bit 0. Bit 0 will be output on only the first iteration of the triggered pattern. Setting bit 0 to an “inactive state” value should minimize or eliminate any ill effects of outputting this bit on the first trigger. Note that on subsequent triggers, there is no additional time latency between trigger and output of bit 1.

4.5.5 Erratum Resolution Status

This erratum is fixed on Apollo4 Plus and Apollo4 Lite SoCs.

4.5.6 AmbiqSuite Workaround Status

A software workaround is not needed in the AmbiqSuite SDK. The user application should take necessary steps to “pad” bit 0 of CMP0 and/or CMP1 with an innocuous value for the pattern of the first trigger, and start the desired output pattern in bit 1.

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

4.6.1 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.

4.6.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.6.3 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.

4.6.4 Workarounds

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

4.6.5 Erratum Resolution Status

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

4.6.6 AmbiqSuite Workaround Status

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

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

4.7.1 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.

4.7.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.7.3 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.

4.7.4 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.

4.7.5 Erratum Resolution Status

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

4.7.6 AmbiqSuite Workaround Status

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

4.8 ERR040: IOS: FIFO read gets stuck/stalled

4.8.1 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.

4.8.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.8.3 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).

4.8.4 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.

4.8.5 Erratum Resolution Status

There are no plans to implement any further fix for this erratum in the Apollo4 family of SoCs.

4.8.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

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

4.9.1 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.

4.9.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.9.3 Application Impact

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

4.9.4 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, the proper system initialization and shutdown sequences minimize the chances of the non-compliant host being affected by the erratum. The proper USB power-up/power-down sequences can be found in the USB section of the Apollo4 SoC Datasheet.

4.9.5 Erratum Resolution Status

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

4.9.6 AmbiqSuite Workaround Status

No workaround is needed in the AmbiqSuite SDK.

4.10 ERR042: MSPI: Violates DMATIMELIMIT0 value if transaction is paused

4.10.1 Description

During an MSPI transaction, the time_limit_hit signal inside the MSPI block indicates when the MSPI unit has hit the time limit indicated by the DMATIMELIMIT0 field of the MSPI DEV0BOUNDARY register. If an MSPI transaction becomes paused, the time_limit_hit signal may not fire at the appropriate time, or at all in some cases.

4.10.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.10.3 Application Impact

User applications may be affected by this erratum when there is system congestion due to other devices (e.g., another MSPI) doing concurrent transfers and competing for the Apollo4 DMA resources.

4.10.4 Workarounds

A workaround for this limitation is to ensure DMA transactions are short thereby reducing the chance that the MSPI will encounter a pause situation.

4.10.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.10.6 AmbiqSuite Workaround Status

There is no explicit software workaround in the AmbiqSuite SDK.

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

4.11.1 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.

4.11.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.11.3 Application Impact

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

4.11.4 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.

4.11.5 Erratum Resolution Status

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

4.11.6 AmbiqSuite Workaround Status

No software workaround is needed in the AmbiqSuite SDK.

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

4.12.1 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.

4.12.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.12.3 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.

4.12.4 Workarounds

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

4.12.5 Erratum Resolution Status

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

4.12.6 AmbiqSuite Workaround Status

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

4.13 ERR049: GPU: GPU hangs when using multiple textures in MRAM

4.13.1 Description

A hang occurs when running the GPU when textures are loaded in MRAM. The issue does not occur when the textures are in SSRAM or external PSRAM. When the hang happens, J-link can attach to the target but the halt command cannot halt the CM4 core. J-link can read the GPU registers, and the GPU status registers shows the value 0xD6800101 or 0xD6800111.

4.13.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.13.3 Application Impact

This issue affects user applications planning to maintain multiple textures in MRAM.

4.13.4 Workarounds

The workaround for this issue is to store the multiple textures in SSRAM or external PSRAM.

4.13.5 Erratum Resolution Status

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

4.13.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a software workaround for this limitation.

4.14 ERR050: MCU_CTRL: High Performance mode entry failure

4.14.1 Description

MCU fails to enter High Performance (192 MHz) mode and code execution hangs after issuing request. After setting PWRCTRL_MCUPERFREQ_MCUPERFREQ = 0x2 to enter HP mode, acknowledgment of entering HP mode is never received resulting in a hang.

4.14.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.14.3 Application Impact

This issue affects user applications by not enabling to go into High Performance mode.

4.14.4 Workarounds

A workaround for this issue exists in the AmbiqSuite SDK - see AmbiqSuite Workaround Status section below.

4.14.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.14.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides a software workaround for this issue in the HAL (am_hal_pwrctrl_mcu_mode_select() function of the PWRCTRL module).

4.15 ERR052: MRAM: MRAM access may be corrupted during Crypto power up/ down or OTP access

4.15.1 Description

Access to MRAM by any master (including CPU) may be corrupted if Crypto is being powered up/down, or if an OTP access is in progress. The MRAM controller does not latch the IFREN signal which is a high-order address bit used to select “INFO Space”. The MRAM controller drives the IFREN signal to the MRAM based on the address being accessed and the device performing the access, such as the ARM M4 or Crypto. It drives other address bits the same way, but other address bits are latched by the macro at the start of the transaction.

Note that Crypto has different ways in which it can access MRAM:

1. It can master the bus and access main MRAM via the same bus interface that the M4 uses to access main MRAM.
2. It can use its own private bus which is connected to an additional port on the MRAM controller used to access the Crypto-only MRAM information known as OTP (IFREN) space.

If an access to OTP from Crypto (number 2 above) overlaps with an access from the M4 to MRAM, then the second transaction (by Crypto) can corrupt the first (M4) transaction by altering the value of IFREN during the Crypto MRAM (OTP) access. This causes incorrect data to be returned to the first accessing device.

The Crypto engine does MRAM accesses during power up while the CM4 is also executing out of MRAM. This defect causes the Crypto to corrupt the CM4’s transactions to MRAM, such that the M4 gets bad data back which can cause various faults (bad instruction, bad literal, etc). In certain conditions, Crypto may not reliably power up after being powered down.

The Crypto does not do accesses to MRAM on power down. However, the defect can cause a spurious transaction, not necessarily caused by the INFREN issue, to MRAM which can corrupt the M4’s access that is in flight.

4.15.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.15.3 Application Impact

This issue affects user applications requiring crypto security.

4.15.4 Workarounds

The workarounds for this issue include the following recommendations as safeguards:

1. Powering up or powering down of Crypto must be done in such a way (with delays) so as to avoid access of MRAM by another master such as DMA, GFX or the Display Controller. This means that DMA cannot access MRAM. A timer can be used to keep the M4 or other masters from doing accesses while Crypto is powering up and accessing MRAM.
2. The M4 accesses the OTP space through the Crypto. The M4 must not access MRAM (i.e., run out of TCM, PSRAM or SSRAM). The SBR and SBL access OTP while running out of MRAM.

3. The defect can cause spurious transaction when MRAM or Crypto is powered off or on. The Crypto must be powered off when entering deep sleep. Also, not turning off MRAM has been shown to prevent the spurious transaction from happening.

4.15.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.15.6 AmbiqSuite Workaround Status

AmbiqSuite SDK enables the following workarounds:

1. Powering up/down of Crypto is done inside code executing from SRAM, which sleeps to allow for all Crypto initiated MRAM transactions to complete.
2. OTP reads & Writes are performed inside a critical section, with code running outside of MRAM, to give exclusive access of MRAM to Crypto
3. Info Reads are run inside a critical section using code running from TCM. [This is a more conservative change.]

Additionally, the application needs to ensure that there are no other masters (CRC, DMA, GPU, etc.) accessing MRAM during Crypto On/Off, or Info/OTP access.

4.16 ERR053: SDIO: Incorrect data is returned on SDIO buffer reads

4.16.1 Description

In some situations, reads from the SDIO interface return incorrect data. Data is transferred over the SDIO interface through the SDIO FIFO. The process for creating synchronous write select (WSEL) bits is missing a step to cause the bits to be explicitly initialized on reset. As a result, the WSEL bits initialize to a random value of 0 or 1 when the design assumes they will be initialized to 0. Any bits which happen to initialize to 1 will result in an error on the first read operation.

If data is written into eMMC for either single block or multiple blocks, but the SoC is power cycled and the card is read without a write, then sometimes the data read is incorrect.

4.16.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.16.3 Application Impact

This issue may affect user applications by not reading eMMC card data reliably.

4.16.4 Workarounds

A workaround is to test the eMMC bus with CMD19 and CMD14 during eMMC initialization and before reading data from eMMC. With CMD19, the dual port RAM in SDIO module will be written, once the dual port RAM is written, it will work as expected. The CMD19 operation has been shown to work with eMMC devices in this case, although other peripheral devices may require a different write operation.

4.16.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.16.6 AmbiqSuite Workaround Status

The above workaround has been added to the eMMC initialization API in the HAL layer of the AmbiqSuite SDK.

4.17 ERR056: USB: High leakage current in USB PHY

4.17.1 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.

4.17.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.17.3 Application Impact

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

4.17.4 Workarounds

A workaround for this issue is to add a weak pull-down resistor (2 Mohm) on both D+ and D-.

4.17.5 Erratum Resolution Status

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

4.17.6 AmbiqSuite Workaround Status

The power sequencing has been updated in dsc_apollo4.c.in SDK version 4.4.0 to better match the power up sequence described in section 18.3.3.1.6 of the Apollo4 SoC datasheets.

4.18 ERR059: TIMER: Functional limitations on all or specific timers

4.18.1 Description

This erratum addresses limitations of functionality and therefore support of the 16 timers (timer 0 - timer 15) in the TIMER module. For certain timers, the degree of non-functionality deems them as not supported for use. These include timers 1, 3 and 10-15, leaving timers 0, 2 and 4-9 usable and supported.

Of the supported timers, use limitations exist as described in Table 3.

Table 3: Timer Use Limitations

Function/Mode	Timer Limitations
TIMER Access	• The TIMER is always clocked in the source clock which is asynchronous to the bus clock. As a result, reads from the TIMER are not guaranteed to be correct. Software should read the TIMER three times in rapid succession (within one period of the source clock), and use the value which is received by at least two of the three reads. • Although it is possible to write the TIMER register, that function will be deprecated in future products and should not be used.
Maximum Source Clock	• The Source Clock selection 0x00 (24 MHz) is deprecated. The maximum allowable frequency is 6 MHz.
AUXEN_TMRnEN Bit Usage	• There is a new control bit in the TIMER_AUXEN register for each TIMER. The TIMER0 enable bit is bit 0 and called TMR00EN, the TIMER1 enable bit is bit 1 and called TMR01EN, and so on. A timer's TMRnEN bit in this register must be set to 1 to reliably enable the timer.
CTRL Register Access	• The CTRLn Register can only be written when either the AUXEN_TMRnEN bit or the CTRL-n_TMRnEN bit is 0. • To use a TIMER, its AUXEN_TMRnEN bit must first be cleared, and then the CTRLn register can be written as needed, e.g., CTRLn_TMRnEN bit is set to 1, etc. Then the AUXEN_TMRnEN is set to 1 to start the TIMER. • During timer operation, any write to the CTRLn register must be preceded by the clearing of the AUXEN_TMRnEN register. • Once timer operation completes, the AUXEN_TMRnEN bit can be cleared and then the CTRL-n_TMRnEN bit can be cleared.
End of Operation Behavior	• The TMRnEN bit may or may not be cleared in hardware at the end of an operation. Software should assume that it is not going to be cleared and clear it manually as described in the next section, using the AUXEN bit. • The TIMER may or may not stop at the end of an operation – software should not assume that it always contains the value of CMP0 at the end. If TMRnEN is not cleared by software within 2^32 source clock cycles, interrupts may be generated again. Note that this is a very long time even for the fastest source clock (at 6 MHz, this is over 10 seconds)
EVENTTIMER Mode Limitations[1]	• The EVENTTIMER mode is unusable.
UPCOUNT/ DOWNCOUNT Mode Limitations[a]	• OUT0 is generated correctly except on the last iteration[2]. OUT0 may or may not be generated on the last iteration.
	• INT0 is generated correctly except on the last iteration. INT0 may or may not be generated on the last iteration.
EDGE Mode Limitations	• OUT0 is unusable.[3] • INT0 is unusable.

Table 3: Timer Use Limitations

Function/Mode	Timer Limitations
PWM Mode Limitations[4]	• OUT0 is generated correctly except on the last iteration (final comparison to CMP0). • OUT1 is generated correctly except on the last iteration (final comparison to CMP0). • INT0 is generated correctly except on the last iteration.
SINGLEPATTERN Mode Limitations[5]	• The pattern length TMRnLMT must be <= 32. • CMP1 should be set to the value which is TMRnLMT – 1. • OUT1 is unusable.
REPEATPATTERN Mode Limitations[6]	• No mode-specific limitations.
Triggering Limitations[7]	• Triggering is not functional on timers 5 and 7-9 of the supported timers.

1. UPCOUNT/DOWNCOUNT Modes: DOWNCOUNT mode works just like UPCOUNT, since they are basically the same (and DOWNCOUNT provides no functionality not provided by UPCOUNT).
2. The “last iteration” is defined to be the one where TMR_LMT is 0x01. This is either the only iteration if TMR_LMT is initialized to 0x01, or the iteration where it has been decremented from 0x02 to 0x01 if TMR_LMT was initialized to a value greater than 0x01. If TMR_LMT is initialized to 0x00, the timer does not terminate and there is no last iteration.
3. EDGE Mode: All other usable TIMERs may be used to generate one-time interrupts by setting CMP1 to the desired time value. CMP0 must be set to a value which is larger than or equal to CMP1, as the operation will stop when CMP0 is reached. Setting CMP0 = CMP1 + 1 is recommended.
4. PWM Mode: Since OUT0 and OUT1 work correctly for each TIMER, a valid PWM signal can be generated for any of the TIMERs except on the last iteration. The TMR_POL0/1 signals work correctly for all TIMERs, so any PWM signal can be created. Depending on the usage of the PWM output, it may be possible to set TMRnLMT to be one higher than the desired num- ber of pulses and terminate the operation when the next to last INT0 is received, thus creating the desired pattern. Since there is no “last iteration” when TMRnLMT is set to 0x00, that operation behaves correctly on at least one output of all TIMERs.
5. SINGLEPATTERN mode: there are some serious issues due to the “last iteration” problem. It is possible that the TIMER does not stop incrementing at the end of the sequence, and in that case the pattern will continue to repeat until software clears the TMRnEN bit. INT1 is generated when the TIMER compares to CMP1, but if CMP1 contains part of the pattern it cannot create a valid INT1. Thus, the only guaranteed functional mode is according to the limitations in the table. Software must be able to service the INT1 interrupt quickly enough to disable the TIMER before the repeated pattern begins. Note that in most cases the TIMER stops correctly, but this cannot be guaranteed.
6. REPEATPATTERN Mode: Since there is no "last iteration" in this mode, operation is as expected on at least one output of all TIMERs. Note that INT1 is generated on the comparison to CMP1. If the content of CMP1 is part of the output pattern (TMRn- LMT > 32), this interrupt is not useful. If TMRnLMT <= 32, OUT0 can be used and CMP1 can be configured to create an INT1 interrupt in each iteration.
7. Triggering: For all timers where triggering works, the trigger selected must come from a TIMER where the selected OUT function is operational.

4.18.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.18.3 Application Impact

The timer issues described in this erratum affect the functionality and use of the timers in user applications in various ways. It may be required to switch timers and their associated pins to avoid limitations of some of the timers (e.g., triggering).

4.18.4 Workarounds

The workaround for this issue is to use the supported timers in a manner that does not violate the use limitations outlined in this erratum.

4.18.5 Erratum Resolution Status

There are no plans to fix these errata in Apollo4 SoC. They are fixed in Apollo4 Plus and Apollo4 Lite.

4.18.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK raised error conditions when a timer is attempted to be configured in a way that is not supported per this erratum.

4.19 ERR063: MSPI: Interface may shut down too early upon RXF condition in non-DQS mode

4.19.1 Description

When using DMA or PIO to receive data transfers to any internal memory with an MSPI interface and when DQS is not in use, the MSPI interface may shut down too early from an internal MSPI FIFO ALMOST FULL (RXF) condition resulting in a missed byte of data. The subsequent data byte will be clocked in twice when the RXF condition is deasserted.

4.19.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.19.3 Application Impact

This issue affects user applications which use DMA or PIO for receiving data transfers on MSPI.

4.19.4 Workarounds

The workaround is to schedule only a single outstanding DMA RX transaction and monitor the INTSTAT_RXF bit to detect a FIFO full condition. The combination of receive target and system load requires monitoring for this condition during in-system testing.

4.19.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.19.6 AmbiqSuite Workaround Status

The specific workaround mentioned above is not implemented in the AmbiqSuite SDK, but the SDK supports detection of the MSPI FIFO FULL condition in the MSPI HAL for non-blocking transactions which use the Command Queue and DMA. The SDK provides examples demonstrating how to use this feature.

4.20 ERR064: IOM: I2C power save/restore failure

4.20.1 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).

4.20.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.20.3 Application Impact

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

4.20.4 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.

4.20.5 Erratum Resolution Status

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

4.20.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK includes changes in the IOM HAL that prevent this issue from occurring.

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

4.21.1 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.

4.21.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.21.3 Application Impact

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

4.21.4 Workarounds

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

4.21.5 Erratum Resolution Status

There are no plans to fix this erratum in any Apollo4 family SoC.

4.21.6 AmbiqSuite Workaround Status

There is no workaround in the SDK for this issue.

4.22 ERR066: IOS: Fails in FIFO mode at specific range of clock frequencies

4.22.1 Description

There is a timing limitation when the IOS is operating in FIFO mode causing the SPI interface to receive 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 occur at interface frequencies between 500 kHz and 15 MHz.

4.22.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.22.3 Application Impact

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

4.22.4 Workarounds

Ensure that the external Master device clocks the IOS in SPI or I2C mode at 15 MHz or higher (SPI only) or 500 kHz or lower (SPI or I2C) when operating in FIFO mode.

4.22.5 Erratum Resolution Status

There are no plans to fix this erratum in any Apollo4 family SoC.

4.22.6 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.

4.23 ERR067: MSPI: RXCAP not operable when CLKDIV=1

4.23.1 Description

The RXCAP0 bit of the MSPI's DEV0CFG register controls the start of the RX data capture phase. A setting of 0 (NORMAL) captures read data on DATA0-DATAn (serial, dual, quad or octal mode) at the normal capture point relative to the internal clock launch point. To accommodate relatively long chip/pad/board delays, an RXCAP setting of 1 delays the data capture phase by one full internal clock cycle (10.4 ns). However, setting RXCAP0 to 1 when DEV0CFG_CLKDIV is set to 1 does not postpone the data capture phase as intended.

4.23.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.23.3 Application Impact

This issue should not affect user applications if other delay fields are used to effectively deal with system delays.

4.23.4 Workarounds

Operating MSPI in a non-DQS configuration when CLKDIV = 1 allows for the setting of the DEV0CFG_RXNEG0 bit to provide a 5 ns delay for read data capture, and for finer tuning by configuring the DEV0DDR_RXDQSDELAY0 field. Also, the DEV0CFG_TURNAROUND0 field allows for adding more delay (10 ns for each step) to increase the time between TX to RX transition, if necessary.

4.23.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.23.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not limit or prevent the configuration of these timing fields when configuring the MSPI interface. User software must configure the interface fields appropriately. In this case setting the RXCAP0 field has no effect.

4.24 ERR070: I2S: I2S0's ASRC Rx is non-operational

4.24.1 Description

Receive capability on the I2S0 instance is functional only when the Asynchronous Sample Rate Converter (ASRC) is disabled. The I2S1 instance is fully functional with or without the ASRC.

4.24.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.24.3 Application Impact

This issue restricts user applications from using I2S0 with ASCR enabled.

4.24.4 Workarounds

The workaround for this issue is to not use I2S instance 0 Rx with ASRC enabled.

4.24.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus.

4.24.6 AmbiqSuite Workaround Status

An AmbiqSuite SDK workaround has been added to am_hal_i2s.c to check the chip revision and ASRC setting, and return an error when using ASRC on I2S0.

4.25 ERR071: INFO: Reads of INFO space fail

4.25.1 Description

Accesses to INFO spaces fail due to a missing latch on the IFREN signal in the 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.

4.25.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.25.3 Application Impact

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

4.25.4 Workarounds

Temporarily program the 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 MRAM controller r_timer1 (which contains Tcycrd) to reset defaults after SBR/SBL but before initial deep sleep entry.

4.25.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.25.6 AmbiqSuite Workaround Status

The device-specific HAL functions for reading INFO space contain the required workarounds, and these HAL functions, e.g., am_hal_mram_info_program() and am_hal_mram_info_read(), should always be used for reading or writing INFO space.

4.26 ERR073: GPU: Blit overruns destination texture

4.26.1 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).

4.26.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.26.3 Application Impact

This issue affects blit display quality in user applications.

4.26.4 Workarounds

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

4.26.5 Erratum Resolution Status

There are no plans to fix this erratum in any Apollo4 family SoC.

4.26.6 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.

4.27 ERR074: DSI: Image displayed outside the set region

4.27.1 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).

4.27.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.27.3 Application Impact

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

4.27.4 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.

4.27.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus.

4.27.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides a software workaround for this issue with the following code in the finish_cmd_transfer() function of nema_dc_hal:

```
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);
```

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

4.28.1 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.

4.28.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.28.3 Application Impact

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

4.28.4 Workarounds

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

4.28.5 Erratum Resolution Status

There are no plans to fix this erratum in any Apollo4 family SoC.

4.28.6 AmbiqSuite Workaround Status

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

4.29 ERR076: MCU_CTRL: Failure to switch from High Performance to Low Power Modes

4.29.1 Description

When switching between HP Mode (CPU running at 192 MHz) and LP Mode (CPU running at 96 MHz), the power controller does not allow enough time for voltage to settle before re-enabling clocks at the new frequency. This, in turn, causes timing issues within the ARM or on its buses and results in a fault due to erroneous fetches or decoding of instructions or addresses.

4.29.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.29.3 Application Impact

This issue may cause MCU/bus execution faults in user applications.

4.29.4 Workarounds

By using sleep mode, the clocks to the ARM can be stopped for a longer period of time while this changeover happens.

4.29.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.29.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK includes the workaround implementation using a TIMER interrupt (TIMER13). The interrupt is configured as the highest priority (0) interrupt to prevent unintentional break out due to other interrupts. In order for this to work reliably, it is required that all other interrupts in the system are set at a lower priority, reserving the highest priority interrupt exclusively for this workaround.

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

4.30.1 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.

4.30.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.30.3 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.

4.30.4 Workarounds

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

4.30.5 Erratum Resolution Status

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

4.30.6 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.

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

4.31.1 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.

4.31.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.31.3 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.

4.31.4 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.

4.31.5 Erratum Resolution Status

There are no plans to fix this erratum in any Apollo4 family SoC.

4.31.6 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.

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

4.32.1 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.

4.32.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.32.3 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.

4.32.4 Workarounds

Enabling SIMO buck can be done in software.

4.32.5 Erratum Resolution Status

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

4.32.6 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).

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

4.33.1 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.

4.33.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.33.3 Application Impact

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

4.33.4 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).

4.33.5 Erratum Resolution Status

There are currently no plans to fix this erratum.

4.33.6 AmbiqSuite Workaround Status

The workaround should be implemented in the user application.

4.34 ERR081: Core: Device hangs when an asynchronous interrupt occurs just after entering deepsleep

4.34.1 Description

The SIMO buck may not come out of low power (LP) mode if a wake from deepsleep interrupt posts about 1.5 us after entering deepsleep. When this happens, the MCU exits deepsleep, switches to the active state, and starts code execution. However, the SIMO buck remains in LP mode and is unable to supply the necessary amount of current to the MCU in active mode, causing the VDDC and VDDF rails to drop out resulting in CPU hang.

4.34.2 Affected Silicon Revisions

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

4.34.3 Application Impact

This issue affects user applications by causing unpredicted hangs if steps are not taken to prevent the hangs.

4.34.4 Workaround

A software workaround is to disable and re-enable the SIMO buck (if the SIMO Buck has been enabled during deep sleep) immediately after deep sleep exit if the SIMO Buck did transition to LP mode during Deep Sleep.

4.34.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.34.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides all functions as shown in the above Workaround section.

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

4.35.1 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).

4.35.2 Affected Silicon Revisions

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

4.35.3 Application Impact

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

4.35.4 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.

4.35.5 Erratum Resolution Status

There are currently no plans to fix this erratum.

4.35.6 AmbiqSuite Workaround Status

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

4.36 ERR083: CLKGEN: HFRC adjustment sometimes does not work

4.36.1 Description

The CLKGEN_HFRCADJ register controls HFRC adjustment logic which is needed when the HFRC clock fluctuates with temperature or needs adjustment due to process variation. The HFRC adjustment logic relies on a divided-down HFRC clock, HFRC_48MHz to work, which requires that HFRC is enabled. The conditions for HFRC being enabled are:

- The MCU is not in deep sleep, AND
- At least one peripheral device is using a clock based on the HFRC clock source OR the CLKGEN_MISC_FRCHFRC bit is enabled OR HFADJ logic is currently adjusting the HFRC.

HFADJ expects that when it requests the HFRC clock, the clock to it is running. However, if no peripherals are requesting a clock, the divider is gated, so HFRC_48MHz is not running and the HFADJ fails.

4.36.2 Affected Silicon Revisions

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

4.36.3 Application Impact

This issue affects user applications by not allowing HFRC adjustment when HFRC_48MHz is not running.

4.36.4 Workarounds

The workaround for this issue is to do the following:

1. Clear CLKGEN_MISC_HFRCFUNCCLKGATEEN.
2. Set the CLKGEN_MISC_CLKGENMISCSPARES field to 0x1e.
3. Clear MISC register, bit 18.

Note that this workaround results in a significant increase in power consumption. A more power-efficient solution is underway which basically generates software timed HFADJ operations rather than the hardware-based adjustments currently implemented. This erratum will be updated when the improved solution is available.

4.36.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.36.6 AmbiqSuite Workaround Status

There is no workaround in the general availability release SDK (R4.0.0). A software workaround is expected in R4.1.0.

4.37 ERR084: TIMER: STIMER capture/compare event cannot be used as a Timer trigger source

4.37.1 Description

TIMER trigger selection, selected by setting the MODEn_TRIGnTRIGSEL field (n = TIMER0 to TIMER15), does not work for any of the STMRCMPn0 or STMRCAPn0 settings (n = STIMER0 to STIMER7), where an STIMER interrupt should trigger the start of a TIMER.

4.37.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.37.3 Application Impact

This issue affects user applications by requiring that a GPIO, which is controlled in an STIMER's ISR, is used to trigger the start of a TIMER.

4.37.4 Workarounds

The workaround for this issue is to set/toggle a GPIO in the STIMER capture/compare ISR, and set this GPIO as the trigger source for the TIMER. This in effect allows the capture or compare function of the STIMER to trigger the start of a TIMER.

4.37.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.37.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK supports the workaround for this issue by allowing the user to use existing SDK functions to configure the workaround.

4.38 ERR085: IOS: Possible failure in FIFO mode in wrap configuration

4.38.1 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.

4.38.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.38.3 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.

4.38.4 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.

4.38.5 Erratum Resolution Status

There are no plans to fix this erratum.

4.38.6 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.

4.39 ERR086: TIMER: Trigger by another timer output doesn't work as intended

4.39.1 Description

When one timer is being triggered by the output of another timer and rising edge or both edge is selected as the trigger edge (CTRLn_TMRnTMODE = RISE or CTRLn_TMRnTMODE = BOTH), and the output polarity of the triggering timer is set such that its output starts high, (CTRLn_TMRnPOLm = 1, where m is OUT0 or OUT1), a false trigger occurs.

4.39.2 Affected Silicon Revisions

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

4.39.3 Application Impact

This issue has minimal affect on user applications if the below workaround is followed. For the use case where a rising edge, or both edges, of another timer output is configured as the trigger source for a timer, the triggering timer's output *cannot* start out high. The timer will start without waiting for the triggering timer's output to traverse low (and then rise to the high state in rising edge trigger mode) to create the edge for the triggered timer to start.

4.39.4 Workarounds

The workaround for this issue is to ensure that the selected timer output for a timer's trigger is low if the configured trigger edge is "rising edge" or "both edges" when the triggered timer is enabled.

4.39.5 Erratum Resolution Status

There are no plans to fix this erratum in Apollo4 SoC. It is fixed in Apollo4 Plus and Apollo4 Lite.

4.39.6 AmbiqSuite Workaround Status

The workaround for this issue is implemented at the application (configuration) level.

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

4.40.1 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.

4.40.2 Affected Silicon Revisions

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

4.40.3 Application Impact

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

4.40.4 Workarounds

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

4.40.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.40.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

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

4.41.1 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.

4.41.2 Affected Silicon Revisions

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

4.41.3 Application Impact

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

4.41.4 Workarounds

NOTE: The below workaround is updated in v13.0 from what was stated previously.

The workaround for this issue is to set both ADC_CALCTRL_ISODLY and ADC_CALCTRL_RESETDLY to 0x3FF (maximum value).

4.41.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.41.6 AmbiqSuite Workaround Status

The HAL in AmbiqSuite SDK R4.5.0 sets ISODLY and RESETDLY to 0x3FF in am_hal_adc_pwrctrl, and removes the former workaround for this issue. With these updated settings, this issue is now resolved in R4.5.0 for all Apollo4 family members.

4.42 ERR091: ADC: Loss of first scan data

4.42.1 Description

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

4.42.2 Affected Silicon Revisions

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

4.42.3 Application Impact

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

4.42.4 Workarounds

NOTE: The below workaround is updated in v13.0 from what was stated previously.

The workaround for this issue is to do the following:

1. Use the only valid ADC clock setting: HFRC_24MHZ.
2. Set both ADC_CALCTRL_ISODLY and ADC_CALCTRL_RESETDLY to 0x3FF (maximum value).

4.42.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.42.6 AmbiqSuite Workaround Status

The HAL in AmbiqSuite SDK v4.5.0 sets ISODLY and RESETDLY to 0x3FF in am_hal_adc_pwrctrl).

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

4.43.1 Description

Use of the DPI-2 interface is not supported.

4.43.2 Affected Silicon Revisions

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

4.43.3 Application Impact

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

4.43.4 Workarounds

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

4.43.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.43.6 AmbiqSuite Workaround Status

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

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

4.44.1 Description

Compare interrupts are delayed by one STIMER clock. Additionally, on Apollo4 Plus and Apollo4 Lite 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.

In addition, back-to-back writes to SCMPRn may not work reliably (i.e., take the last value) on Apollo4 Plus and Apollo4 Lite 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.

It takes two STIMER clock cycles for the write to STCFG to take effect. This caused 2 extra cycles to add to the minimum delta.

4.44.2 Affected Silicon Revisions

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

4.44.3 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.

4.44.4 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 SoC and Apollo4 Lite SoC, this latency correction is 3. Therefore, the minimum valid delta setting is 4. For Apollo4, the latency correction is 1, resulting in a minimum valid delta setting of 2.

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.

4.44.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.44.6 AmbiqSuite Workaround Status

The AmbiqSuite HAL contains the workaround for these delays starting in SDK release 4.5.0 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 SoC and Apollo4 Lite SoC, 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. The application needs to check for and handle such a case.

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

4.45.1 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).

4.45.2 Affected Silicon Revisions

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

4.45.3 Application Impact

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

4.45.4 Workarounds

There is no workaround for this issue.

4.45.5 Erratum Resolution Status

There are no plans at this time to fix this erratum on Apollo4 family SoCs.

4.45.6 AmbiqSuite Workaround Status

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

4.46 ERR100: TIMER: High Deep Sleep current when TMR6/TMR9 enabled

4.46.1 Description

There is significantly higher current when running timers 6 or 9 in Deep Sleep than when using other timers.

4.46.2 Affected Silicon Revisions

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

4.46.3 Application Impact

This issue affects user applications by higher current draw in deep sleep mode when either of these two timers is used.

4.46.4 Workarounds

The workaround for this issue is to use another timer if the timer is going to operate in Deep Sleep, or use STIMER if it provides needed functionality.

Note that when any timer (or any other peripheral) selects an HFRC-based clock, the HFRC remains running in Deep Sleep. This produces a significant increase in Deep Sleep current independent of the timer selected. Enabling multiple timers in this case has a small effect on current, as the HFRC itself is where nearly all of the current is used.

4.46.5 Erratum Resolution Status

There currently are no plans to fix this erratum on Apollo4 SoC. It has been fixed on Apollo4 Plus and Apollo4 Lite.

4.46.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK allows for the selection and use of any of the supported timers. The application should not use TIMER6 or TIMER9 if it is to be used in Deep Sleep mode.

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

4.47.1 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.

4.47.2 Affected Silicon Revisions

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

4.47.3 Application Impact

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

4.47.4 Workarounds

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

4.47.5 Erratum Resolution Status

There are no plans at this time to fix this erratum on Apollo4 family SoCs.

4.47.6 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.

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

4.48.1 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 SWFLAG2 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.

4.48.2 Affected Silicon Revisions

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

4.48.3 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.

4.48.4 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.

4.48.5 Erratum Resolution Status

There currently are no plans to fix this erratum on Apollo4 family SoCs.

4.48.6 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.

4.49 ERR103: IOM: FIFO threshold interrupt incorrectly triggered

4.49.1 Description

In normal read operation, the FIFO threshold 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 threshold 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.

4.49.2 Affected Silicon Revisions

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

4.49.3 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.

4.49.4 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.

4.49.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.49.6 AmbiqSuite Workaround Status

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

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

4.50.1 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.

4.50.2 Affected Silicon Revisions

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

4.50.3 Application Impact

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

4.50.4 Workarounds

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

4.50.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.50.6 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.

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

4.51.1 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.

4.51.2 Affected Silicon Revisions

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

4.51.3 Application Impact

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

4.51.4 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.

4.51.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.51.6 AmbiqSuite Workaround Status

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

4.52 ERR107: RTC: Clock domain crossing issue causes APB bus hang

4.52.1 Description

In extremely rare occurrences, an APB bus hang may occur when doing a read or write access to the RTC's CTRUP or CTRLOW registers due to insufficient synchronization logic in the RTC block. The unlikely occurrence of the failure has been observed in about 20% of devices tested. However among those failing devices, the rate of occurrence of a single failure varies but is generally in the range of millions of counter register reads, which translates to thousands of hours of operation in a typical RTC use case. No other triggers of this failure have been identified.

4.52.2 Affected Silicon Revisions

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

4.52.3 Application Impact

This issue adversely affects user applications by losing control over functionality of modules when an APB hang occurs, requiring some type of reset, e.g., watchdog reset, to recover.

4.52.4 Workarounds

Reducing the number of CTRUP and CTRLOW register accesses proportionately reduces the risk of encountering the failure. Also, a read of the RTCSTAT register before accessing the CTRUP and CTRLOW registers significantly reduces the rate of occurrence of the APB lock up.

A watchdog reset may be implemented to effectively recovery from such a failure.

4.52.5 Erratum Resolution Status

There currently are no plans to fix this erratum on Apollo4 SoC. It has been fixed on Apollo4 Plus and Apollo4 Lite.

4.52.6 AmbiqSuite Workaround Status

There currently is no software workaround implemented in the AmbiqSuite SDK for this issue although reading the RTCSTAT register before accessing the RTC counter registers to reduce the failure rate is easily performed.

4.53 ERR108: MSPI: Timing issue using CLKOND4

4.53.1 Description

An internal timing issue causes unreliable data reads when using data line D4 for the MSPI's clock (MSPIn_PADOUTEN_CLKOND4 = 1). The option to use D4 as the clock line has therefore been deprecated for all MSPI instances of the Apollo4 SoC. The MSPI's D8 line should be used as the clock line for all data widths.

4.53.2 Affected Silicon Revisions

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

4.53.3 Application Impact

This issue affects user applications only by restricting the options for the clock line.

4.53.4 Workarounds

The workaround for this issue is to use the default clock line, MSPIn_D8.

4.53.5 Erratum Resolution Status

There currently are no plans to fix this erratum. The use of the CLKOND4 setting to use data line 4 as the clock has been deprecated on Apollo4 SoC.

4.53.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK supports the use of MSPIn_D8 for the clock line.

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

4.54.1 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.

4.54.2 Affected Silicon Revisions

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

4.54.3 Application Impact

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

4.54.4 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.

4.54.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.54.6 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.

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

4.55.1 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.

4.55.2 Affected Silicon Revisions

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

4.55.3 Application Impact

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

4.55.4 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.

4.55.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.55.6 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.

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

4.56.1 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.

4.56.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.56.3 Application Impact

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

4.56.4 Workarounds

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

4.56.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.56.6 AmbiqSuite Workaround Status

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

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

4.57.1 Description

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

4.57.2 Affected Silicon Revisions

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

4.57.3 Application Impact

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

4.57.4 Workarounds

NOTE: The below workaround is updated in v13.0 from what was stated previously.

The workaround for this issue is to do the following:

1. Use the only valid ADC clock setting of 24 MHz (ADC_CFG_CLKSEL = HFRC_24MHZ).

NOTE: Because of this erratum, 48 MHz ADC clock is no longer supported on any Apollo4 Family SoC. The ADC_CFG_CLKSEL field selections have been updated to indicate that HFRC_24MHZ is the only valid selection.

4.57.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.57.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK HAL allows the selection of the HFRC_24MHZ clock source in support of the specified workaround.

4.58 ERR114: BLE: Performance not guaranteed during concurrent USB operation

4.58.1 Description

Due to interference between USB clock circuitry and the BLE Controller, BLE performance is not guaranteed during simultaneous USB operation. The PLL of the USB module outputs a 480 MHz clock and, when the USB and BLE operate concurrently, the clock's high-order harmonics interfere with the bandwidth of the BLE and affects the radio frequency characteristics of certain BLE channels.

4.58.2 Affected Silicon Revisions

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

4.58.3 Application Impact

This issue affects concurrent BLE and USB operation in user applications.

4.58.4 Workarounds

The use of the USB module is supported only for firmware updates, debugger I/O and serial logging output. One workaround for this issue is to not enable and use the BLE during these limited USB operations if at all possible.

If concurrent USB and BLE operation is necessary, then the influence of the PLL on the BLE can be avoided by adjusting the clock accuracy (negative bias) of HFRC since the reference clock of the USB PLL comes from HFRC.

By default, the output of the HFRC is 96 MHz. The impact of the USB on BLE operation can be avoided by using the function of MCU HFADJ and adjusting the output of HFADJ to 95 MHz by configuring the HFADJ register. This workaround has been shown to effectively avoid the influence of the USB PLL on BLE radio frequency performance through verification in high and low temperature environments.

The software enables HFADJ and configures the HFRC output to 95 MHz with the following code:

```
uint32_t ui32Regval =
_VAL2FLD(CLKGEN_HFADJ_HFADJMAXDELTA, 14) |                  /* Max Adjust step size set 14*/
_VAL2FLD(CLKGEN_HFADJ_HFADJGAIN,
    CLKGEN_HFADJ_HFADJGAIN_Gain_of_1) |                     /* HF Adjust with Gain of 1 */
_VAL2FLD(CLKGEN_HFADJ_HFWARMUP,
    CLKGEN_HFADJ_HFWARMUP_1SEC) |                           / * Default value */
_VAL2FLD(CLKGEN_HFADJ_HFXTADJ, 0x5AC) |                     / * HFRC output ~95M */
_VAL2FLD(CLKGEN_HFADJ_HFADJCK,
    CLKGEN_HFADJ_HFADJCK_4SEC) |                            / * Default value */
_VAL2FLD(CLKGEN_HFADJ_HFADJEN, CLKGEN_HFADJ_HFADJEN_EN);
ui32Regval |= _VAL2FLD(CLKGEN_HFADJ_HFADJEN,
```

```
    CLKGEN_HFADJ_HFADJEN_EN);                    /* HFADJ Enable */
CLKGEN->HFADJ = ui32Regval;
```

NOTE: Users should be mindful when utilizing this workaround that all modules that use HFRC will now operate at the reduced frequency. Users should verify that this meets the requirements of their system.

4.58.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.58.6 AmbiqSuite Workaround Status

Use of the BLE and USB modules is controlled in the user application. The AmbiqSuite SDK provides enablement and initialization functions for each of these modules.

4.59 ERR115: GPIO: Possible glitch on GPIO outputs upon initial power up

4.59.1 Description

There is a possibility of experiencing a glitch on GPIO outputs due to the initial state of an internal level shifter being bi-stable during power up when those outputs are temporarily pulled up to VDD_MCU.

4.59.2 Affected Silicon Revisions

This issue is applicable to affected GPIO on all silicon revisions of Apollo4 SoC.

4.59.3 Application Impact

The effect of this issue is dependent upon the sensitivity to a glitch on the circuit/peripheral that is connected to the GPIO. It is likely to not have a detrimental effect.

4.59.4 Workarounds

There currently is no workaround for this issue.

4.59.5 Erratum Resolution Status

There currently are no plans to fix this erratum on Apollo4 SoC. It has been fixed on Apollo4 Plus and Apollo4 Lite.

4.59.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a software workaround for this issue.

4.60 ERR118: DC: Cannot read data from display panel in SPI mode

4.60.1 Description

In SPI, DualSPI or QuadSPI mode, the Display Controller (DC) cannot read data from a display panel. the DC can correctly write to a panel in each SPI modes. This issue does not affect other DC interface modes.

4.60.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.60.3 Application Impact

This issue prevents reading data from the display panel in a standard SPI interface.

4.60.4 Workarounds

A workaround for not being able to perform a DC-SPI read from the display panel is to enable the DBI Type-B interface before the read, use DC-SPI interface for the read, and then disable DBI Type-B interface after the read. The DBI Type-B interface can be enabled by setting the DBITYPEBEN bit in the DC's Mode register. For the purpose of reading a panel, the DBI Type B interface width can be set to 8-, 9- or 16-bits by configuring the TYPEBWIDTH field of the DC’s DBICFG register. DBI mode can be disabled after the read by clearing the DBITYPEBEN bit and reconfiguring for the desired SPI mode.

4.60.5 Erratum Resolution Status

This erratum is intended to be fixed on a future SoC family.

4.60.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a software workaround for this issue, but does support the setup and use of the DBI Type-B interface. See the am_devices_display_generic.c and .h files for display mode functions.

4.61 ERR119: ADC: Incorrect sample rate when using Internal ADC Timer as repeat clock source

4.61.1 Description

Using the internal ADC timer as the repeating trigger clock source (ADC_CFG_RPTTRIGSEL = INT), there are two additional, unexpected clock cycles added to the total divide cycle. This means the ADC sample rate will be decreased by two 24 MHz clock cycles.

4.61.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.61.3 Application Impact

This issue affects user applications by using an incorrect sample rate when the internal ADC timer is used as the periodic trigger clock source.

4.61.4 Workarounds

Workarounds for this issue are the following:

1. When using the Internal Repeating Trigger Timer, verify that two additional 24 MHz clock cycles in this sample clock generator will not impact the application. If it does, either configure the ADC_CLKDIV_-TIMERMAX to take into account these two extra clock cycles, or:
2. Use Timer 7 as the periodic trigger clock source (ADC_CFG_RPTTRIGSEL = TMR, and TIMER_-GLOBEN_ADCEN = EN).

4.61.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.61.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a workaround for this erratum. It is up to the application to take into account these two extra clock cycles if it is a concern.

4.62 ERR121: CLKGEN: XTAL32K is activated when XTAL_HS is selected as module clock

4.62.1 Description

The high-speed crystal clock, referred to as XTHS or XTALHS, is offered as a module clock option (CLKSEL) for the AUDADC, PDM and I2S modules. However, this clock selection operation has the unintended side effect of enabling the 32 kHz crystal at the same time.

4.62.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.62.3 Application Impact

Whenever the high-speed crystal is being used by the AUDADC, PDM or I2S module in active mode or in any low-power mode, the 32 kHz crystal will be enabled and drawing power.

4.62.4 Workarounds

There is no workaround for the 32 kHz crystal being powered and active when the high-speed crystal is selected as the module clock for the AUDADC, PDM or I2S module. To avoid this situation, when possible, clock these modules with one of the other clock options.

4.62.5 Erratum Resolution Status

This erratum is intended to be fixed in a future SoC family.

4.62.6 AmbiqSuite Workaround Status

The HAL function, am_hal_mcuctrl_control, in the AmbiqSuite SDK implements the proper way to set up the high-speed crystal in preparation for selecting it as the clock source for the AUDADC, PDM and I2S modules.

4.63 ERR123: TIMER: Timer module reset required between HW trigger-initiated timer operations

4.63.1 Description

When using a HW trigger to initiate timer start in any mode on any timer, a specific trigger, selected by MODEn_TMRnTRIGSEL, can be used only once before requiring a reset of the Timer module in order to use the trigger source again. A Timer module reset stops all existing timer operations of all timers and requires the reconfiguration and restart of all desired timer operations.

4.63.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.63.3 Application Impact

Since this issue requires a timer module reset and re-initialization of any running timers between sequential trigger-initiated operations, there is timer set-up overhead incurred in user applications. As well, there is a discontinuation of any other concurrent timer operations since the workaround for this issue requires a complete timer module reset (to use the hardware trigger again).

4.63.4 Workarounds

The only way to cause the (re)triggering of a timer operation or a different operation requiring a trigger after a completed triggered-timer operation is to reset the entire Timer module (CTRL_RESET= 1). This terminates all existing timer operations and requires that timers be reconfigured and restarted. If a HW trigger is not used, then the single timer can be reconfigured and operation can be re-initiated as normal and with no disruption to other timer operations.

A software workaround is to use a high-priority interrupt (Timer or GPIO) to generate the desired trigger. Within the ISR of that interrupt, use software commands to start the timer. The penalty in using this workaround is that the ISR causes a slight delay in timer execution/output and requires CPU intervention to run the ISR resulting in a slight cost in power draw.

4.63.5 Erratum Resolution Status

There currently are no plans to fix this erratum for this device. Note that on the Apollo4 Lite and Apollo4 Blue Lite SoCs, this triggering issue is fixed and does not require the resetting of the entire Timer module.

4.63.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides the Timer reset function as well as other functions necessary to configure for all supported timer operations.

4.64 ERR124: MSPI: Mixed Mode 1-1-4 does not work as expected

4.64.1 Description

A glitch occurs on the D0 line between address and data in mixed mode D4, or 1-1-4. The glitch is observed only during write operation in 1-1-4 mode and its occurrence may cause data write failures. Even though glitches have been observed on all Apollo4 family devices, data write failures have only been observed on the Apollo4 SoC. Read operation is not affected and therefore it does not pose a problem for XIP/XIPMM reads.

4.64.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.64.3 Application Impact

This issue may cause write data errors in MSPI D4 mode when the glitch is sampled by an external device.

4.64.4 Workarounds

A workaround for this issue is to emulate D4 mode in software. Disable the ADDR phase (DEV0XIP_XIPSENDA = 0), then add extra data at the beginning of the data phase to emulate D4 mode.

There should be no problem for XIP/XIPMM reads but there is no workaround for XIPMM write issue in 1-1-4 mode. Note that XIP/XIPMM across 1-1-4 interface is not a common use case since it is normally used in NAND flash which does not support XIP/XIPMM.

4.64.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.64.6 AmbiqSuite Workaround Status

The workaround code is released in the NAND flash driver of the AmbiqSuite SDK.

4.65 ERR125: BLE: Corrupted non-volatile memory prevents boot-up of the BLE controller

4.65.1 Description

In a low percentage of devices, malfunction of the BLE (Bluetooth Low Energy) controller occurs after multiple power cycles. The issue is caused by the BLE controller power supply (VDDB) ramping up too quickly which in turn may cause erroneous erasure of the BLE controller firmware within its non-volatile memory. This erasure prevents the BLE controller from booting up correctly and therefore cannot function and is unrecoverable.

4.65.2 Affected Silicon Revisions

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

4.65.3 Application Impact

This issue affects user applications by causing the BLE controller to function incorrectly or not at all.

4.65.4 Workarounds

Low failure rate is observed when the VDDB power-up ramp rate (the time for VDDB to transition from 0 V to 1.8 V) is less than 100 µs, and is significantly reduced or eliminated at a ramp rate longer than 350 µs. Depending on the system configuration, slowing down the VDDB power-up ramp rate could be achieved by one of the following methods:

- Adjusting the external PMIC's slew
- Increasing the external PMIC's or VDDB's bypass cap to slow the rise time
- Adding a simple transistor soft start / slow turn-on circuit in series with the VDDB pin

4.65.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.65.6 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 with the use of the described workaround.

4.66 ERR126: CLKGEN: HFADJ enabled with no HFRC-clocked modules powered causes incorrect HFRC clock frequency

4.66.1 Description

The internal HFRC Adjust (HFADJ) circuit incorrectly uses the gated instead of the ungated HFRC clock. When there are no peripherals enabled and actively being clocked by HFRC, the HFRC clock used by HFADJ is gated which causes the HFADJ oscillator to select an out-of-spec high frequency for the adjustment of HFRC. When an HFRC-clocked peripheral is then enabled, it will be clocked at an out-of-spec high clock rate which may result in unpredictable register initialization and peripheral operation.

In order to keep HFRC from being gated at HFADJ, at least one peripheral must have HFRC selected as its clock source and also the peripheral must drive the clock enable high (i.e., the peripheral needs to be actually using the clock). Any peripheral that can be clocked by HFRC can be used for this purpose.

4.66.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.66.3 Application Impact

An incorrectly clocked module may cause the module to not operate correctly if at all.

4.66.4 Workarounds

To avoid this erratum and incorrect HFRC clock rate, ensure that HFRC adjust (HFADJ) is disabled when no clients are using it, and enable it only after a peripheral has been powered on and started being clocked by HFRC.

4.66.5 Erratum Resolution Status

There currently are no plans to fix this erratum on Apollo4 SoC. It has been fixed on Apollo4 Plus and Apollo4 Lite SoCs.

4.66.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK for the Apollo4 SoC has been updated as follows:

1. Upon CLKGEN HFADJ Enable, if no powered modules, return error.
2. Add 10 µs delay after enable/disable to allow clocks to settle.
3. If the HFADJ was enabled directly with no peripheral enabled, then disable HFADJ, enable first peripheral, and re-enable HFADJ.

4.67 ERR127: BLE: Image corruption at boot or reset

4.67.1 Description

Due to the absence of a pull-down on the BLE controller reset line within an Apollo4 family “Blue” SoC, the radio can end up in an indeterminate state at boot-up or reset. When the BLE Controller secure bootloader (SBL) attempts to authenticate the controller firmware image while in an uncertain state, an authentication error may occur. When an authentication failure occurs, the SBL attempts to write a failure flash to the controller flash. If the reset pin or voltage rail status is uncertain, it is possible for the flash write to result in flash memory corruption.

4.67.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.67.3 Application Impact

If the authentication failure flag is written or if the BLE controller flash is corrupted, the BLE controller is no longer functional and becomes unrecoverable.

4.67.4 Workarounds

The BLE Controller SBL has been updated to no longer write to flash when the SBL image authentication fails. This eliminates the possibility of having a permanent authentication failure, and mitigates the possibility of flash corruption.

In addition, the BLE Controller reset pin can be controlled by software to minimize the amount of time the controller is in an unknown state.

4.67.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.67.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK has been updated to add additional BLE Controller reset line control at power-up in am_bsp_low_power_init() for each EVB, as well as when the radio is shut down in hci_drv_cooper.c.

4.68 ERR128: MSPI: D3:D1 lines are pulled low instead of staying in high impedance mode

4.68.1 Description

Behavior of the upper data lines (IO2 and IO3) during the instruction phase of mixed-mode (1-1-4 or 1-4-4) transfers is preventing IO3 to function properly as the nHOLD signal, as is done on certain NAND devices. When nHOLD is pulled low at the start of the transaction and despite pull-ups present, it causes the chip to ignore all other signals. The SoC should leave the upper data lines floating during the instruction phase of mixed-mode transfers.

4.68.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.68.3 Application Impact

Certain NAND devices requiring IO3 to be kept high during mixed-mode (1-1-4 or 1-4-4) operation may not function properly.

4.68.4 Workarounds

Currently there is no workaround for this issue. Use of certain NAND devices requiring IO3 to be kept high during mixed-mode (1-1-4 or 1-4-4) operation is not recommended or supported.

4.68.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.68.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide a solution for, and does not support, the use of these devices.

4.69 ERR129: RTC: CB field value is unpredictable when year = 99 and CEB = 1.

4.69.1 Description

When the RTC_CTRUP_CEB bit is set to enable the Century bit (CB) to change, the CB toggles whenever and as long as the Year field (CTRYR) is 99. This happens not just on rollover from 99 to 00 but toggles every time the 10 ms counter changes, which is (essentially) every RTC clock.

4.69.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.69.3 Application Impact

This issue may affect any application using POSIX time which uses January 1, 1970 as its epoch. In such systems a common test involves checking the correct value of the Century bit upon rollover from1999 to 2000.

4.69.4 Workarounds

The workaround for this issue is to handle the unpredictability of the CB bit when the year is 99 by either ignoring the Century bit (CB) setting in year 99 when the CEB field is set to 1, or do not set the CEB bit keeping the CB static.

4.69.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.69.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK has been modified as follows:

- CEB is no longer supported.
- CB is used to determine 20xx or 21xx only; 19xx has been removed.
- No rollover from Year 99 to Year 00.

4.70 ERR130: CLOCKGEN: HF2ADJ-introduced jitter may cause incorrect HFRC2 adjustment

4.70.1 Description

When HF2ADJ is enabled, the HFRC2 adjustment circuit may incorrectly adjust the HFRC2 output due to jitter being introduced by the adjustment circuit.

4.70.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 SoC.

4.70.3 Application Impact

The out-of-range adjustments mentioned above can result in clocking issues with USB high-speed and other blocks clocked by HFRC2.

4.70.4 Workarounds

A workaround for this issue is to periodically run a HF2ADJ tuning procedure and average the HF2ADJ trimming output to adjust HFRC2 sufficiently without significantly increasing jitter.The HF2ADJ tuning procedure is summarized in the following sequence.

HFRC2 Adjust (HF2ADJ) Initial Setup Procedure:

1. Ensure that the 32 MHz high speed crystal oscillator (XTHS) is running.
2. Disable HFRC2 FLL by setting CLKGEN_HF2ADJ0_HF2ADJEN = DIS.
3. Set the HF2ADJ trimming offset value in the CLKGEN_HF2ADJ1_HF2ADJTRIMOFFSET field to 0 (default). If user wants to modify this signed 11-bit number, that is permissible.
4. The HF2ADJ output selection (CLKGEN_HF2ADJ1_HF2ADJTRIMEN field) sets the output mux and must be set to enable the FLL (bit 0 set), therefore any field option that has bit 0 set (e.g., TRIM_EN1, TRIM_EN5, etc.) such that the output of the FLL is actually used to generate the HFRC2 frequency. If set to TRIM_EN5 the following values are used to generate the HFRC2 output frequency:
   A. HF2ADJTRIMOUT (output of the FLL)
   B. HF2TUNE (HFRC2 tune value set in the MCUCTRL_HFRC2_HF2TUNE field)
5. Select the clock multiplier/divider for the FLL.
   A. The input clock for Apollo4 is 32 MHz. This is run through a pre-divider to reduce the frequency below 12 MHz.
   B. Choose a divider of 4, so the FLL input freq is 8 MHz. In this case the intermediate FLL frequency (IF) will be target-output-freq * 16. For an FLL output target of 24 MHz that IF is 384 MHz.
   C. For an 8 MHz base FLL input clock, the multiplier to 384 is 48, that is scaled by 2^{15}, so the register value used for 24MHz is: Ratio_value =48 * 2^{15} = 0x180000.
   D. Set CLKGEN_HF2ADJ2_HF2ADJRATIO = Ratio_value.
6. Start the HFRC2 FLL (HF2ADJ) by setting CLKGEN_HF2ADJ0_HF2ADJEN = EN.
7. Check the chip revision number (MCUCTRL_CHIPREV) and/or patch configuration to determine if the MCUCTRL_HFRC2_HF2TUNE field is visible to the processor. If the chip does not support HFRC2 tune, leave the HFRC2 FLL running (Pi controller).

8. Run the following repeating tune procedure for the first time.

Repeating Tune Procedure:

1. Set the trim value to TRIM_EN5 for the HF2ADJ output selection (CLKGEN_HF2ADJ1_HF2ADJTRIMEN field).
2. Enable the 32 MHz high speed crystal oscillator (XTHS) if not running. The 32 Mhz clock takes about 300 µs to start up, so delay for this duration after enabling.
3. Enable the HFRC2 FLL by setting CLKGEN_HF2ADJ0_HF2ADJEN = EN.
4. Wait for the FLL to stabilize (about 1000 µs delay).
5. Average the output of the FLL. Fetch data from the HF2ADJ trimming output (CLKGEN_HF2VAL_HF2ADJTRIMOUT field).
   A. The data is 11-bit signed.
   B. Mask the 11 bits and sign extend to 32-bit signed.
   C. Sample data no faster than 1 µs per sample.
   D. Average (sum) the data.
   E. This averaged data must be converted back to 11-bit signed.
6. Load the averaged value into MUCCTRL_HFRC2_HF2TUNE.
7. Disable the HFRC2 FLL by setting CLKGEN_HF2ADJ0_HF2ADJEN = DIS.
8. Disable the 32 MHz XTHS (if desired).

Repeat the Tune procedure approximately every 10 seconds to compensate for temperature variation.

4.70.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.70.6 AmbiqSuite Workaround Status

The HFRC2 tuning procedure described in the Workaround above is implemented as examples in the AmbiqSuite SDK starting with version 4.5.0:

- power/pwr_32Mhz
- usb/tinyusb_cdc_hfrc2
- usb/tinyusb_cdc_msc_hfrc2

These examples use HFRC2 adjust (HF2ADJ) to create a 24 MHz HFRC2 frequency and calls the function *am_hal_clkgen_HFRC2_adj_recompute()* every 10 seconds.

The function *am_hal_clkgen_HFRC2_adj_recompute()* executes the tune procedure every ten seconds.

5. Ordering Information

Table 4: SoC Ordering Information

Device Name	Orderable Part Number[a]	MRAM	RAM	Package (mm)	Packing	Temperature Range
Apollo4 SoC	AMAP42KK-KBR	2 MB	1.8 MB	5.0 x 5.0 146-pin BGA	Tape and Reel	–20 to 60°C
Apollo4 Blue SoC	AMA4B2KK-KBR	2 MB	1.8 MB	4.7 x 4.7 131-pin BGA	Tape and Reel	–20 to 60°C

a. The silicon revision is identified by the first letter in the bottom row of the package's top marking.

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-A4-13p0
Version 13.0
June 2024