ERRATA LIST

Apollo4 Lite SoC, Apollo4 Blue Lite SoC

Ultra-low Power Apollo SoC Family

Doc. ID: SE-A4L-4p0

Doc. Revision: 4.0, 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

Silicon Errata for the **Apollo4 Lite SoC and Apollo4 Blue Lite SoC**

1. Introduction

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

2. Document Revision History

Table 1: Document Revision History

Rev No.	Date	Description
1.0	Feb 2023	Initial Release
2.0	Jul 2023	Updated errata: • ERR071: Removed from list as was fixed on this device. Added errata: • ERR116, ERR117, ERR120 - ERR122, ERR124
3.0	Feb 2024	Updated errata: • ERR002: Updated Erratum Resolution Status • ERR003: Updated Description and Erratum Resolution Status • ERR040: Updated Erratum Resolution Status • ERR064: Updated AmbiqSuite Workaround Status • ERR065: Updated Erratum Resolution Status • ERR066: Updated Description, Resolution and Erratum Resolution Status fields. • ERR073: Updated Erratum Resolution Status • ERR075: Updated Erratum Resolution Status • ERR078: Updated Erratum Resolution Status • ERR080: Updated Erratum Resolution Status • ERR091: Updated Workarounds section. • ERR098: Clarified Description, Workarounds and AmbiqSuite Workaround Status fields. • ERR099: Updated Erratum Resolution Status • ERR101: Updated Erratum Resolution Status • ERR105: Updated Description • ERR113: Removed extraneous text at end of page. Added errata: • ERR125

Table 1: Document Revision History

Rev No.	Date	Description
4.0	June4.0 2024	Removed errata: • ERR117: This erratum limiting UART2 to 24 MHz has been removed; all UART instances now limited to 24 MHz by specification. 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, ERR127 - 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 Lite SoC revision as well unless otherwise stated.

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
"ERR002: ADC: DMA ISR cannot be properly triggered" on page 10	All existing	No Fix planned	Software workaround
"ERR003: CACHE: Some modes are unsupported" on page 11	All existing	Partially fixed; no further fix planned.	No workaround
"ERR038: IOM: CQ fails to complete DMA read transfers" on page 12	All existing	No fix planned	Software workaround
"ERR039: IOS: MISO line is not tri-stated when CE driven high" on page 13	All existing	No fix planned	Software/hardware workaround
"ERR040: IOS: FIFO read gets stuck/stalled" on page 14	All existing	Partially fixed; no further fix planned.	No workaround
"ERR043: DMA: Incorrect reads/writes near memory boundaries" on page 15	All existing	No fix planned	Software workaround
"ERR046: GPIO: FIEN/FOEN not operational on GPIO0" on page 16	All existing	No fix planned	Software/hardware workaround
"ERR064: IOM: I2C power save/restore failure" on page 17	All existing	No fix planned	Software workaround
"ERR065: MSPI: CM4 hard fault not triggered when it should be" on page 18	All existing	No fix planned	Software workaround
"ERR066: IOS: Fails in FIFO mode at specific range of clock frequencies" on page 19	All existing	No fix planned	No workaround
"ERR073: GPU: Blit overruns destination texture" on page 20	All existing	No fix planned	Software workaround
"ERR075: I2S: Clocks incorrectly gated while in deep sleep" on page 21	All existing	No fix planned	Limited software workaround
"ERR078: MSPI: Potential race condition when using RXNEG and RXDQSDELAY concurrently" on page 22	All existing	No fix planned	Software workaround
"ERR079: INFO0: SIMO Buck cannot be enabled via INFO0 setting" on page 23	All existing	No fix planned	Software workaround
"ERR080: MSPI: Command Queue may disable DMAEN while data is still pending in internal buffer" on page 24	All existing	No fix planned	Software workaround
"ERR085: IOS: Possible failure in FIFO mode in wrap configuration" on page 25	All existing	No fix planned	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
"ERR087: MCU_CTRL: POR failure due to VDDC/ VDDF not rising to proper level" on page 26	All existing	No fix planned	Hardware workaround
"ERR088: SDIO: DDR mode not supported" on page 27	All existing	No fix planned	No workaround
"ERR090: ADC: No CNVCMP interrupt for first single scan" on page 28	All existing	No fix planned	Software workaround
"ERR091: ADC: Loss of first scan data" on page 29	All existing	No fix planned	Software workaround
"ERR097: MSPI: Non-DQS SDR Octal not reliably operable at 48 MHz or 96 MHz" on page 30	All existing	No fix planned	Software workaround
"ERR098: STIMER: Constraints on writing to SCMPRn registers and handling Compare interrupts" on page 31	All existing	No fix planned	Software workaround
"ERR099: IOM: CQ does not pause via the BLE module" on page 33	All existing	No fix planned	No workaround
"ERR101: IOM: Command write causes CQ operations to pause and never restart" on page 34	All existing	No fix planned	Software workaround
"ERR102: IOM: CQ does not pause immediately after triggering event" on page 35	All existing	No fix planned	Software workaround
"ERR103: IOM: FIFO threshold interrupt incorrectly triggered" on page 36	All existing	No fix planned	Software workaround
"ERR104: IOM: Data corrupted on I2C when OFFSETCNT=0 and I2CLSB=1" on page 37	All existing	No fix planned	Software workaround
"ERR105: GPU: System hang when GPU fetches data from TCM" on page 38	All existing	No fix planned	Software workaround
"ERR110: DAXI: Out-of-order SSRAM read and write returns incorrect read value" on page 39	All existing	No fix planned	Software workaround
"ERR111: MSPI: Delayed MSPI write b-response may cause MSPI state machine deadlock" on page 41	All existing	No fix planned	Software workaround
"ERR112: ADC: Dummy trigger causes immediate (invalid) interrupt" on page 43	All existing	No fix planned	Software workaround
"ERR113: ADC: Occasional corrupt conversion results at 48 MHz" on page 44	All existing	No fix planned	Software workaround
"ERR116: DEBUG: Cannot wake up the MCU with DAP" on page 45	All existing	No fix planned	Software workaround
"ERR119: ADC: Incorrect sample rate when using Internal ADC Timer as repeat clock source" on page 46	All existing	No fix planned	Software workaround
"ERR120: GPU: Possibility of hang during GPU power down" on page 47	All existing	No fix planned	Software workaround
"ERR121: CLKGEN: XTAL32K is activated when XTAL_HS is selected as module clock" on page 49	All existing	No fix planned	No workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
"ERR122: BootROM: Incorrect total RAM size used for source address" on page 50	All existing	No fix planned	Software workaround
"ERR124: MSPI: Mixed Mode 1-1-4 does not work as expected" on page 52	All existing	No fix planned	Software workaround
"ERR125: BLE: Corrupted non-volatile memory prevents boot-up of the BLE controller" on page 53	Blue versions only	No fix planned	Hardware workaround
"ERR127: BLE: Image corruption at boot or reset" on page 54	All existing	No fix planned	Software workaround
"ERR128: MSPI: D3:D1 lines are pulled low instead of staying in high impedance mode" on page 55	All existing	No fix planned	Hardware workaround
"ERR129: RTC: CB field value is unpredictable when year = 99 and CEB = 1." on page 56	All existing	No fix planned	Software workaround
"ERR130: CLOCKGEN: HF2ADJ-introduced jitter may cause incorrect HFRC2 adjustment" on page 57	All existing	No fix planned	Software workaround

4. Detailed Silicon Errata

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

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

4.1 ERR002: ADC: DMA ISR cannot be properly triggered

4.1.1 Description

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

Note that this issue applies only to the ADC on Apollo4 Lite since there is no AUDADC on this device.

4.1.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo4 Lite 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 Lite 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 ERR038: IOM: CQ fails to complete DMA read transfers

4.3.1 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.

4.3.2 Affected Silicon Revisions

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

4.3.3 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.

4.3.4 Workarounds

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

4.3.5 Erratum Resolution Status

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

4.3.6 AmbiqSuite Workaround Status

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

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

4.4.1 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.

4.4.2 Affected Silicon Revisions

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

4.4.3 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.

4.4.4 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.

4.4.5 Erratum Resolution Status

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

4.4.6 AmbiqSuite Workaround Status

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

4.5 ERR040: IOS: FIFO read gets stuck/stalled

4.5.1 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.

4.5.2 Affected Silicon Revisions

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

4.5.3 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).

4.5.4 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.

4.5.5 Erratum Resolution Status

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

4.5.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

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

4.6.1 Description

The APBDMA reads or writes unexpected values near the boundary of target memories when the target DMA start address is not 32-byte aligned. This issue may occur when DMA transactions straddle the SRAM (TCM) and SSRAM boundary.

4.6.2 Affected Silicon Revisions

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

4.6.3 Application Impact

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

4.6.4 Workarounds

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

4.6.5 Erratum Resolution Status

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

4.6.6 AmbiqSuite Workaround Status

No software workaround is needed in the AmbiqSuite SDK.

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

4.7.1 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.

4.7.2 Affected Silicon Revisions

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

4.7.3 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.

4.7.4 Workarounds

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

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 software workaround in the AmbiqSuite SDK related to this loss of functionality on GPIO0.

4.8 ERR064: IOM: I2C power save/restore failure

4.8.1 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).

4.8.2 Affected Silicon Revisions

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

4.8.3 Application Impact

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

4.8.4 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.

4.8.5 Erratum Resolution Status

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

4.8.6 AmbiqSuite Workaround Status

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

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

4.9.1 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.

4.9.2 Affected Silicon Revisions

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

4.9.3 Application Impact

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

4.9.4 Workarounds

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

4.9.5 Erratum Resolution Status

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

4.9.6 AmbiqSuite Workaround Status

There is no workaround in the SDK for this issue.

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

4.10.1 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.

4.10.2 Affected Silicon Revisions

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

4.10.3 Application Impact

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

4.10.4 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.

4.10.5 Erratum Resolution Status

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

4.10.6 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.

4.11 ERR073: GPU: Blit overruns destination texture

4.11.1 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).

4.11.2 Affected Silicon Revisions

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

4.11.3 Application Impact

This issue affects blit display quality in user applications.

4.11.4 Workarounds

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

4.11.5 Erratum Resolution Status

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

4.11.6 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.

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

4.12.1 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.

4.12.2 Affected Silicon Revisions

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

4.12.3 Application Impact

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

4.12.4 Workarounds

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

4.12.5 Erratum Resolution Status

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

4.12.6 AmbiqSuite Workaround Status

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

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

4.13.1 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.

4.13.2 Affected Silicon Revisions

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

4.13.3 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.

4.13.4 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.

4.13.5 Erratum Resolution Status

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

4.13.6 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.

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

4.14.1 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.

4.14.2 Affected Silicon Revisions

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

4.14.3 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.

4.14.4 Workarounds

Enabling SIMO buck can be done in software.

4.14.5 Erratum Resolution Status

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

4.14.6 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).

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

4.15.1 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.

4.15.2 Affected Silicon Revisions

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

4.15.3 Application Impact

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

4.15.4 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).

4.15.5 Erratum Resolution Status

There are currently no plans to fix this erratum.

4.15.6 AmbiqSuite Workaround Status

The workaround should be implemented in the user application.

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

4.16.1 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.

4.16.2 Affected Silicon Revisions

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

4.16.3 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.

4.16.4 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.

4.16.5 Erratum Resolution Status

There are no plans to fix this erratum.

4.16.6 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.

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

4.17.1 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.

4.17.2 Affected Silicon Revisions

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

4.17.3 Application Impact

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

4.17.4 Workarounds

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

4.17.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.17.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

4.18 ERR088: SDIO: DDR mode not supported

4.18.1 Description

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

4.18.2 Affected Silicon Revisions

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

4.18.3 Application Impact

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

4.18.4 Workarounds

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

4.18.5 Erratum Resolution Status

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

4.18.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

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

4.19.1 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.

4.19.2 Affected Silicon Revisions

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

4.19.3 Application Impact

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

4.19.4 Workarounds

NOTE: The below workaround is updated in v4.0 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).

4.19.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.19.6 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.

4.20 ERR091: ADC: Loss of first scan data

4.20.1 Description

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

4.20.2 Affected Silicon Revisions

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

4.20.3 Application Impact

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

4.20.4 Workarounds

NOTE: The below workaround is updated in v4.0 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).

4.20.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.20.6 AmbiqSuite Workaround Status

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

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

4.21.1 Description

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

4.21.2 Affected Silicon Revisions

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

4.21.3 Application Impact

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

4.21.4 Workarounds

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

4.21.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.21.6 AmbiqSuite Workaround Status

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

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

4.22.1 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 four for Apollo4 Lite.

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.

4.22.2 Affected Silicon Revisions

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

4.22.3 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.

4.22.4 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.

4.22.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.22.6 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.

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

4.23.1 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).

4.23.2 Affected Silicon Revisions

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

4.23.3 Application Impact

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

4.23.4 Workarounds

There is no workaround for this issue.

4.23.5 Erratum Resolution Status

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

4.23.6 AmbiqSuite Workaround Status

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

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

4.24.1 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.

4.24.2 Affected Silicon Revisions

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

4.24.3 Application Impact

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

4.24.4 Workarounds

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

4.24.5 Erratum Resolution Status

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

4.24.6 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.

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

4.25.1 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.

4.25.2 Affected Silicon Revisions

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

4.25.3 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.

4.25.4 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.

4.25.5 Erratum Resolution Status

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

4.25.6 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.

4.26 ERR103: IOM: FIFO threshold interrupt incorrectly triggered

4.26.1 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.

4.26.2 Affected Silicon Revisions

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

4.26.3 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.

4.26.4 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.

4.26.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.26.6 AmbiqSuite Workaround Status

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

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

4.27.1 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.

4.27.2 Affected Silicon Revisions

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

4.27.3 Application Impact

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

4.27.4 Workarounds

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

4.27.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.27.6 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.

4.28 ERR105: GPU: System hang when GPU fetches data from TCM

4.28.1 Description

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

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

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

4.28.2 Affected Silicon Revisions

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

4.28.3 Application Impact

This issue adversely affects user applications if enabling the GPU to try to access data from TCM.

4.28.4 Workarounds

The workaround for this issue requires two steps:

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

4.28.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.28.6 AmbiqSuite Workaround Status

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

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

4.29.1 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.

4.29.2 Affected Silicon Revisions

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

4.29.3 Application Impact

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

4.29.4 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.

4.29.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.29.6 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.

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

4.30.1 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.

4.30.2 Affected Silicon Revisions

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

4.30.3 Application Impact

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

4.30.4 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.

4.30.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.30.6 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.

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

4.31.1 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.

4.31.2 Affected Silicon Revisions

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

4.31.3 Application Impact

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

4.31.4 Workarounds

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

4.31.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.31.6 AmbiqSuite Workaround Status

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

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

4.32.1 Description

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

4.32.2 Affected Silicon Revisions

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

4.32.3 Application Impact

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

4.32.4 Workarounds

NOTE: The below workaround is updated in v4.0 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.

4.32.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.32.6 AmbiqSuite Workaround Status

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

4.33 ERR116: DEBUG: Cannot wake up the MCU with DAP

4.33.1 Description

When the MCU is in Deep Sleep mode and is connected to the Debug Access Port (DAP), the MCU cannot be woke up by the DAP.

4.33.2 Affected Silicon Revisions

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

4.33.3 Application Impact

Not allowing the debugger to wake the MCU when in Deep Sleep mode prevents the ability to diagnose and debug user applications when in this mode.

4.33.4 Workarounds

The workaround for this issue is to enter burst mode before entering deepsleep mode. The DAP is able to wake up the MCU using this setting.

4.33.5 Erratum Resolution Status

There are no plans to fix this issue on Apollo4 Lite SoC.

4.33.6 AmbiqSuite Workaround Status

The above workaround is implemented for Apollo4 Lite in SDK R4.4.0.

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

4.34.1 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.

4.34.2 Affected Silicon Revisions

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

4.34.3 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.

4.34.4 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).

4.34.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.34.6 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.

4.35 ERR120: GPU: Possibility of hang during GPU power down

4.35.1 Description

An MCU hang may occur during the power down of the GPU module. During GPU power down, it is possible to have an inadvertent transaction on the primary AXI fabric, in particular, those targeting MRAM. After the GPU powers down, the MRAM return data for the transaction cannot be acknowledged causing the MRAM interface to remain in a busy state. Any subsequent access to MRAM (primarily from the CPU) will not be serviced causing the CPU to stall.

4.35.2 Affected Silicon Revisions

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

4.35.3 Application Impact

This issue may occur in a user application, possibly causing the MCU to hang requiring power cycling or system reset, i.e., external pin reset or watchdog reset.

4.35.4 Workarounds

To prevent this issue from occurring, an incorrect AXI transaction needs to be cleared before the next request from the CPU to MRAM. This requires the following:

- The CPU must run from TCM memory to prevent any MRAM transaction from occurring until the AXI transaction is cleared.
- Use WFI in deep sleep to ensure that an AXI transaction gets properly cleared which resets the fabric without side effects.
- No AXI transactions must occur during the WFI window as this will prevent deep sleep entry and an AXI transaction to clear.
- No pending unmasked interrupts must occur as this will prevent deep sleep entry.

Upon WFI exit, any incorrect AXI transaction will have been cleared and then normal accesses can resume.

4.35.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.35.6 AmbiqSuite Workaround Status

Software has been updated in the 4.4.0 SDK and certain code should be implemented in customer software as follows:

- SDK changes:
  - Drivers for all DMA-enabled peripherals have been updated to prevent any AXI transactions to occur during the WFI window. Files that include some of these drivers are am_hal_mspi.c, am_hal_iom.c and am_hal_pdm.

  - The main workaround has been implemented in nema_hal.c, located in third_party\ThinkSi\config\apollo4l_nemagfx.
- Customer software:
  - Link File changes should be made to reserve 20 kB of TCM to put critical files needed by the workaround there to avoid MRAM access.

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

4.36.1 Description

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

4.36.2 Affected Silicon Revisions

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

4.36.3 Application Impact

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

4.36.4 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 PDM or I2S module. To avoid this situation, when possible, clock these modules with one of the other clock options.

4.36.5 Erratum Resolution Status

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

4.36.6 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 PDM and I2S modules.

4.37 ERR122: BootROM: Incorrect total RAM size used for source address

4.37.1 Description

For Apollo4 Plus, the total size of RAM should be 384+1024+384+1024=2816 kB, or a last valid word-aligned source address of 0x102BFFFC (valid only if writing exactly one word). However, the total size of SRAM/SSRAM used is the same size as Apollo4, which is a maximum of 384+1024+480=1888 kB, resulting in a last valid source address of 0x101D7FFC.

This results in the Apollo4 Plus BootROM helper functions limiting the source memory range to 0x10000000 - 0x101D7FFC, the same as would be valid for Apollo4. Any calling function attempting to use a valid Apollo4 Plus source address of 0x101D8000 - 0x102BFFFC will fail with return code 4.

The issue also affects Apollo4 Lite since it uses the exact same BootROM binary, but affects it in a different way. That is, the Apollo4 Lite RAM should be sized as 384+1024=1408 kB, or a valid address range of 0x10000000 - 0x1015FFFC. However, the BootROM helper functions will accept source addresses up to 0x101D7FFC. So the helper functions will fail in an undetermined way, possibly even a hang situation, if an invalid source address is used.

4.37.2 Affected Silicon Revisions

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

4.37.3 Application Impact

This issue affects user applications or factory production programs which use certain BootROM helper functions using source buffers located in SRAM. On Apollo4 Plus, this issue limits the usable range of RAM addresses passed to a BootROM helper function. The issue affects Apollo4 Lite user applications in an undetermined manner if an out-of-bounds source address is attempted to be written by a helper function.

The helper functions affected are the following:

- nv_program_main()
- nv_program_main2()
- nv_program_info_area()
- nv_program_main_from_sram()
- nv_program_main2_from_sram()
- nv_program_info_area_from_sram()

4.37.4 Workarounds

The workaround for this issue on Apollo4 Plus and Apollo4 LIte is to limit the range of RAM-based source buffers passed to any of the affected BootROM helper functions (as listed in the description) as follows:

- For Apollo4 Plus, the workaround is to ensure that no part of a RAM-based source buffer exceeds the range 0x10000000 - 0x101D7FFC. Doing so will result in an error code of 4 being returned to the caller.
- For Apollo4 Lite, the workaround is to ensure that no part of a RAM-based source buffer exceeds the range of 0x10000000 - 0x1015FFFC, that is, the valid RAM address space for Apollo4 Lite. Doing so will result in undefined behavior.

4.37.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.37.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides validation of the source addresses for functions am_hal_mram_main_words_program() and am_hal_mram_info_program(). These are not workarounds, but simply checks of the buffer address range. It is up to the user application to ensure that the source buffer passed directly to a BootROM helper function is valid.

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

4.38.1 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.

4.38.2 Affected Silicon Revisions

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

4.38.3 Application Impact

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

4.38.4 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.

4.38.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.38.6 AmbiqSuite Workaround Status

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

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

4.39.1 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.

4.39.2 Affected Silicon Revisions

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

4.39.3 Application Impact

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

4.39.4 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

4.39.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.39.6 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.

4.40 ERR127: BLE: Image corruption at boot or reset

4.40.1 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.

4.40.2 Affected Silicon Revisions

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

4.40.3 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.

4.40.4 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.

4.40.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.40.6 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.

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

4.41.1 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.

4.41.2 Affected Silicon Revisions

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

4.41.3 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.

4.41.4 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.

4.41.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.41.6 AmbiqSuite Workaround Status

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

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

4.42.1 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.

4.42.2 Affected Silicon Revisions

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

4.42.3 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.

4.42.4 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.

4.42.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.42.6 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.

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

4.43.1 Description

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

4.43.2 Affected Silicon Revisions

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

4.43.3 Application Impact

The out-of-range adjustments mentioned above can result in clocking issues with blocks clocked by HFRC2.

4.43.4 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.

4.43.5 Erratum Resolution Status

Currently there are no plans to fix this erratum.

4.43.6 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 3: SoC Ordering Information

Device Name	Orderable Part Number[a]	MRAM	RAM	Package (mm)	Packing	Temperature Range
Apollo4 Lite SoC	AMAP42KL-KBR	2 MB	1.375 MB	5.0 x 5.0 146-pin BGA	Tape and Reel	–20 to 60°C
Apollo4 Blue Lite SoC	AMA4B2KL-KXR	2 MB	1.375 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-A4L-4p0
Version 4.0
June 2024