ERRATA LIST

Apollo510 SoC / Apollo510B SoC

Ultra-low Power Apollo SoC Family

Doc. ID: SE-A510-3p0

Doc. Revision: 3.0, December 2025

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 **Apollo510 SoC** and **Apollo510B SoC**

1. Introduction

This document is a detailed compilation of known device errata for the general availability revisions, B1 and B2, of the Apollo510 SoC and Apollo510B SoC. Unless stated otherwise, all listed errata apply to both SoCs and all packages of the SoCs.

2. Document Revision History

Table 1: Document Revision History

Rev No.	Date	Description
1.0	December 2024	Initial Release
2.0	February 2025	ERR024 updated: AmbiqSuite Workaround Status expanded. ERR028 updated: AmbiqSuite Workaround Status corrected. ERR033, ERR034, ERR036, ERR037 added.
3.0	December 2025	Added errata information for Apollo510B SoC. ERR001 updated: Included Apollo510B-specific information in Description. ERR002 updated: Workarounds corrected. ERR030 updated: Included Apollo510B-specific information. ERR038 - ERR046 added.

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.

NOTE: Since errata may or may not be applicable to all devices of an SoC family, errata numbering in the list for a particular device may have gaps.

Table 2: Errata Summary

Erratum Number, Title and Page	Apollo510 SoC	Apollo510B SoC	Affected Silicon Revisions	Resolution Status	Workaround
“ERR001: AUDADC: MCU hangs when attempting to configure the AUDADC with disabled clock” on page 9	X	X	All existing	No fix planned	Software workaround
“ERR002: DC: Limitations with MRAM AXI bus read transactions” on page 11	X	X	All existing	Fix on future derivatives	Software workaround
“ERR003: DEBUG: In self-hosted debug, TPIU register writes when PDDBG is not power up can cause system hang” on page 12	X	X	All existing	No fix planned	Software workaround
“ERR005: DSI: Cannot complete DBI transaction when reading DSI register” on page 13	X		All existing	No fix planned	Software workaround
“ERR006: DSI: DSI PHY draws 8 mA when powered up” on page 14	X	X	Revision B1	Fixed on revision B2	Software workaround
“ERR007: RTC: Intermittent bus hang by reading either the CTRUP or CTRLOW registers in LP or HP mode” on page 15	X	X	All existing	No fix planned	Software workaround
“ERR008: IOM: FIFO threshold interrupt incorrectly triggered” on page 16	X	X	All existing	No fix planned	Software workaround
“ERR009: MSPI: Accessing some powered-down/unclocked memory interfaces can cause system hangs” on page 17	X	X	All existing	No fix planned	Software workaround
“ERR010: IOS: FIFO read gets stuck/stalled” on page 19	X		All existing	No fix planned	No workaround
“ERR011: IOS: MISO line is not tri-stated when CE is de-asserted” on page 20	X	X	All existing	No fix planned	Hardware and Software workarounds
“ERR012: IOS: FIFO mode failures with certain FIFO_BASE and FIFOPTR settings” on page 21	X		All existing	No fix planned	Software workaround
“ERR013: IOS: Subordinate interrupt (SLINT) is not bonded out to external pin” on page 22	X		All existing	No fix planned	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Apollo510 SoC	Apollo510B SoC	Affected Silicon Revisions	Resolution Status	Workaround
“ERR014: MEMORY: Restrictions on use of DMA to TCM” on page 23	X	X	All existing	Fix on future derivatives	Software workaround
“ERR015: MEMORY: SIMO Buck cannot be enabled via INFO0 setting” on page 24	X	X	All existing	No fix planned	Software workaround
“ERR016: MSPI: Peripheral accesses not guaranteed when clock source <= 48 MHz” on page 25	X	X	All existing	No fix planned	No workaround
“ERR018: MSPI: Crossing a memory boundary during DMA transfer causes device hang” on page 26	X	X	All existing	No fix planned	Software workaround
“ERR019: MSPI: Upper data lines are pulled low instead of staying in high impedance mode” on page 27	X	X	All existing	No fix planned	No workaround
“ERR020: MSPI: Mixed Mode 1-1-4 does not work as expected” on page 28	X	X	All existing	No fix planned	Software workaround
“ERR021: PDM: High OSR causes aliasing and PDM mic distortion to occur during the PDM-to-PCM conversion” on page 29	X	X	All existing	Fix on future derivatives	Software workaround
“ERR022: PWRCTRL: APB bus hang when accessing unpowered OTP registers or INFO space” on page 30	X	X	All existing	Fix on future derivatives	No workaround
“ERR023: PWRCTRL: CPU cannot go into deepsleep when either OTP or ROM is powered on” on page 31	X	X	All existing	Fix on future derivatives	Software workaround
“ERR024: PWRCTRL: MCU continues to use the clock for LP mode (HFRC) after switching from LP to HP mode” on page 32	X	X	All existing	No fix planned	Software workaround
“ERR025: PWRCTRL: SIMO Buck fails to transition from Active Mode to LP Mode” on page 33	X	X	All existing	No fix planned	Software workaround
“ERR026: PWRCTRL: Some power domains might not power up as expected after a SWPOI reset” on page 34	X	X	All existing	Fix on future derivatives	Software workaround
“ERR027: RSTGEN: Brown-out reset status bit may get set inadvertently during power-up” on page 35	X	X	All existing	No fix planned	Software workaround
“ERR028: RTC: CB field value is unpredictable when year = 99 and CEB = 1” on page 36	X	X	All existing	No fix planned	Software workaround
“ERR029: STIMER: Constraints on writing to SCMPRn registers and handling Compare interrupts” on page 37	X	X	All existing	No fix planned	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Apollo510 SoC	Apollo510B SoC	Affected Silicon Revisions	Resolution Status	Workaround
“ERR030: SYSPLL: Special case where EXTREFCLK cannot be set as the SYSPLL clock” on page 39	X	X	All existing	Fix on future derivatives	Software workaround
“ERR031: USB: Induced D+ output pulse may cause unintended disconnect” on page 41	X	X	All existing	No fix planned	No workaround
“ERR032: DEBUG: SWO pin goes low during deepsleep” on page 42	X	X	All existing	Fix on future derivatives	Software workaround
“ERR033: Boot Loader: Keybank restrictions when used for OTA encryption” on page 43	X	X	All existing	Fix on future derivatives	Software workaround
“ERR034: Boot Loader: SBL's OEM recovery does not use INFO0 settings for wired update” on page 44	X	X	All existing	Fixed in SBL version 1.16	No workaround
“ERR036: Secure Boot ROM: SDIO1 can only be used in 1-bit mode for MRAM recovery” on page 45	X	X	All existing	No fix planned	Hardware and software workaround
“ERR037: IOS/IOSFD: Erroneous XCMPRF interrupt triggered” on page 46	X	X	All existing	No fix planned	Software workaround
“ERR038: BootROM: Clock for MSPI used for MRAM recovery not selectable” on page 47	X	X	All existing	No fix planned	Software workaround
“ERR039: DC: Incorrect pixel data output order from the DC SPI” on page 48	X	X	All existing	No fix planned	Software workaround
“ERR040: DEBUG: Cannot connect to a debugger after DEBUG domain is powered down” on page 49	X	X	All existing	No fix planned	Software workaround
“ERR041: DEBUG: DEBUG power domain may not power down under certain conditions” on page 50	X	X	All existing	No fix planned	Software workaround
“ERR042: I2S: Error flag is not set when a DMA error occurs” on page 51	X	X	All existing	No fix planned	Software workaround
“ERR043: MRAM: Concurrent AXI burst read to MRAM and an MRAM write cause read data to be corrupted” on page 52	X	X	All existing	No fix planned	Software workaround
“ERR044: MRAM: Powered down NVM1 causes state machine to hang” on page 53	X	X	All existing	No fix planned	Software workaround
“ERR045: TMR: Timer does not start counting until 4th clock tick” on page 54	X	X	All existing	No fix planned	Software workaround
“ERR046: USB: High leakage current in USB PHY” on page 55	X	X	All existing	No fix planned	Hardware 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 ERR001: AUDADC: MCU hangs when attempting to configure the AUDADC with disabled clock

4.1.1 Description

If an AUDADC register access is attempted when the clock source connection has been removed, the MCU will hang. If the AUDADC is on and any loss-of-clock condition occurs, this can result in a CPU hang. The loss-of-clock condition could be one of the following:

- Disabling XTAL_HS while the AUDADC is being clocked by it. Note that XTAL_HS is not a valid selection for, and therefore this condition should never exist on, the Apollo510B SoC.
- When the AUDADC is clocked by EXTREFCLK on GPIO15 and its pad function changes from EXTREFCLK to another preventing the clock input, or the clock connection is removed physically, or the external clock source itself stops providing the clock.
- Selecting “OFF” for the AUDADC clock source.
- Disabling SYSPLL when the AUDADC is clocked by the SYSPLL.
- Disabling the high frequency crystal oscillator output when the AUDADC is clocked by the PLL and the high frequency crystal oscillator is the SYSPLL reference clock. Note that the high frequency crystal oscillator output is not a valid selection for, and therefore this condition should never exist on, the Apollo510B SoC.
- Disabling EXTREFCLK when the AUDADC is clocked by the PLL and EXTREFCLK is SYSPLL reference clock.

Also, powering on the AUDADC and accessing an AUDADC register after a software reset, but the clock source before the reset has not been enabled after the reset, an AUDADC register access will cause an MCU hang to occur.

4.1.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.1.3 Application Impact

This issue may affect user applications that clock the AUDADC module with the high-speed crystal (XTAL_HS), the SYSPLL or by an external clock (EXTREFCLK).

4.1.4 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 HFRC_48MHz.
2. Before powering down the AUDADC, set the module’s clock to an HFRC/HFRC2 selection if the clock selection was the XTAL_HS, SYSPLL or EXTREFCLK.

Then there should be no problem when the AUDADC is next initialized and powered on. The module’s clock source may then be reselected upon re-enabling the AUDADC, making sure that the intended clock is present and enabled.

1. The firmware should not set the Clock Select to OFF for AUDADC and should not disable the direct/indirect clock source(s) currently used by the AUDADC.

2. When using EXTREFCLK, be sure not to reconfigure GPIO15 to another function, or stop the external clock source when the AUDADC module is powered on, or assign an alternate function to GPIO15 altogether.

This ensures that the clock source for AUDADC is always active and the CPU hang issue doesn't occur during device operation.

4.1.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.1.6 AmbiqSuite Workaround Status

The Ambiqsuite SDK invokes am_hal_sysctrl_clkmuxrst_low_power_init() in the am_hal_pwrctrl_low_power_init() function, which sets the clock to HFRC_48MHz to prevent the hang when enabling the AUDADC.

NOTE: Clock source selection for the AUDADC is managed by the AUDADC HAL and the Clock Manager, and the user does not need to worry about the clock configuration. So enabling XTAL_HS no longer requires the user application to do it - it is done by the AUDADC/I2S/PDM HAL by calling the Clock Manager. The Clock Manager will not turn off a clock requested by AUDADC as long as it is not released by AUDADC as well.

4.2 ERR002: DC: Limitations with MRAM AXI bus read transactions

4.2.1 Description

When the Display Controller (DC) is fetching data from multiple memories for its multiple layers and one of the memories is MRAM, failures occur in the form of incorrect data being read or an AXI bus hang.

4.2.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.2.3 Application Impact

This issue may cause AXI bus hang.

4.2.4 Workarounds

A workaround for this issue is to not use MRAM as the source for the DC, but instead use a memory such as SSRAM or external pSRAM

4.2.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 devices. This erratum is intended to be fixed in future derivatives of the Apollo5 family.

4.2.6 AmbiqSuite Workaround Status

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

4.3 ERR003: DEBUG: In self-hosted debug, TPIU register writes when PDDBG is not power up can cause system hang

4.3.1 Description

On the M55, self-hosted debug components should be powered on automatically when the M55 writes to any of the debug subsystem registers (ITM, WDT, TPIU, etc.). In the Apollo510, writing to the TPIU registers does not automatically power-on the debug subsystem. Therefore software must manually enable the debug power domain, PDDBG, by setting the PWRCTRL_DEVPWREN_PWRENDBG bit, and then can access the debug registers safely. This is required before accessing any self-hosted debug functions, including enabling the SWO for logging or messages. If software writes to a TPIU register when the PDDBG is off, the system hangs. Note that manually enabling PDDBG is not required when an external debugger is attached.

4.3.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.3.3 Application Impact

This issue may cause a CPU hang if the PDDBG power domain is not powered up before writing to debug registers.

4.3.4 Workarounds

A workaround for this issue is to explicitly enable the PDDBG power domain before writing the TPIU registers. If the PDDBG is off, then the self-hosted wakeup may fail. Self-hosted debug means that it is only the CPU accessing the ITM/TPIU, and this workaround is still required. If the TPIU is programmed just for SWO, then the software workaround is to enable the PDDBG when in use and then manually power down the PDDBG.

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’s HAL routines, such as am_hal_tpiu_enable() and am_hal_itm_enable(), and their complementary functions am_hal_itm_disable() and am_hal_tpiu_disable(), explicitly handle the power-up and power-down of the debug domain.

4.4 ERR005: DSI: Cannot complete DBI transaction when reading DSI register

4.4.1 Description

Cannot complete the DBI transaction when reading a DSI display register and the 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 the 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 an ACK from the display. It will enter a busy state, and will not set DBI_DataValid. As a result the DBI cannot be freed and 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 the DBI_DV (DBI_DataValid) signal must also be used to terminate the read transaction.

4.4.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 SoC.

4.4.3 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.

4.4.4 Workarounds

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

4.4.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.4.6 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.

4.5 ERR006: DSI: DSI PHY draws 8 mA when powered up

4.5.1 Description

When the DSI PHY is powered up, a VSS power switch produces an additional 8 mA of current leakage. The VSS power switch should be opened when DSI PHY is powered up but there is a PN diode in the PMOS switch that creates a forward bias current from VDDF to VSS.

4.5.2 Affected Silicon Revisions

This silicon erratum applies to all “B” revisions of Apollo510 SoC up to revision B1. It is fixed on revision B2.

4.5.3 Application Impact

This issue causes an additional 8 mA current draw when the DSI PHY is powered on and being used due to leakage.

4.5.4 Workarounds

There currently is no workaround for this issue.

4.5.5 Erratum Resolution Status

This erratum is fixed in silicon revision B2.

4.5.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK cannot provide a workaround to prevent the extra current draw.

4.6 ERR007: RTC: Intermittent bus hang by reading either the CTRUP or CTRLOW registers in LP or HP mode

4.6.1 Description

An intermittent bus hang may result from reading either the RTC's CTRUP or CTRLOW register in Low Power (LP) or High Performance (HP) mode. When the read access occurs close to the rising edge of the 100 Hz RTC clock, the read finite state machine (FSM) could potentially transition to an unknown state and get stuck there causing a bus hang. Only a reset returns the FSM back to normal.

4.6.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.6.3 Application Impact

Under certain timing conditions, this issue may cause a bus hang recoverable only by reset.

4.6.4 Workarounds

A workaround for this issue is to avoid reading the two RTC registers close to the rising edge of the RTC clock. The workaround can either leverage the system timer (STIMER) provided by the application to minimize the power impact, or use the reserved timer in the HAL to capture the rising edge. Setting the read delay to around 10 µs is adequate. If the CTRUP/CTRLO register reads are NOT within the RTC's ISR which would occur coincident with the clock's rising edge, then an additional delay waiting for the next rising edge would need to be incurred.

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 has the workaround mentioned above implemented.

4.7 ERR008: IOM: FIFO threshold interrupt incorrectly triggered

4.7.1 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 doing a FIFO read transaction (DMA or non-DMA), the write threshold check is not gated off. This could trigger an incorrect/invalid write interrupt which should be ignored. If the application uses the wrong interrupt to check the read data, incorrect data processing could result.

4.7.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.7.3 Application Impact

The effect of this issue on user applications depends on how the application handles IOM interrupts and whether FIFO pointers are evaluated properly before initiating a FIFO read operation to determine how much data is ready to be read. If an invalid write interrupt is received and handled as a read command completed and read data is ready to use, then processing wrong data could result.

4.7.4 Workarounds

The workaround for this issue is for the application to do the following:

1. To prevent the triggering of a wrong interrupt, the FIFOWTHR should be set to zero when performing a read command and the FIFORTHR should be set to zero when performing a write command.
2. Always inspect the actual FIFO pointers before determining how much data is available to read. If the application inspects the actual FIFO pointer properly, then errant (false threshold) interrupts can be ignored and there should not be any adverse issues due to this erratum.

4.7.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.7.6 AmbiqSuite Workaround Status

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

4.8 ERR009: MSPI: Accessing some powered-down/unclocked memory interfaces can cause system hangs

4.8.1 Description

An explicit memory access (read, write and instruction fetch) or a data and instruction pre-fetch to a memory that is in a powered down or unclocked state may cause a system hang. If there is dirty data in the data cache, a cacheable access to another data region may cause a cache-line eviction when attempting to write back the cache line to memory. If that memory is powered down, then a system hang may occur.

The following table lists access types and resulting error codes/conditions for accesses to either SSRAM, MSPI or MSPI's PSRAM memory.

Access Type by CPU	SSRAM	MSPI/PSRAM
Memory Access in Retention/Standby	SSRAM is out of retention if CPU is active. Auto-Wakeup occurs for accesses from other mains.	MSPI is on. Software can put PSRAM in Standby / low power mode. MSPI does not know the PSRAM is in a low-power state, so it will continue to process read/write requests which may get no response and may cause a system hang.
Memory Access if powered down	RAXI returns SLVERR for all mains.	System hang occurs.[1]
Memory access if powered up but not clocked	There is no SSRAM clock disable.	System hang occurs.
Register Access in Retention	Registers in MCUCTRL and PWRCTRL are always on.	MSPI is still on even if the PSRAM is put into a low power mode. MSPI register accesses will work correctly.
Register Access if power off	Registers are in MCUCTRL and PWRCTRL and are always on.	Decoder will detect and return a SLVERR.
Register Access if power on but not clocked	Registers in MCUCTRL and PWRCTRL are always on.	Decoder will not block the access, so system hang occurs.
Protection violation	Protection violation will be raised, no access will be generated on the system fabric.	Protection violation will be raised, no access will be generated on the system fabric.

1. When the M55 attempts to access a powered down MSPI device, the MSPI device cannot respond to the request. The system does not detect this condition, so an error is not returned. This results in unsatisfied memory transactions that can hang the system. This issue is unique to MSPI, but applies to all 4 MSPI instances. Accesses to other powered down memories (SSRAM, DTCM, etc) will return the expected memory fault.

4.8.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs.

4.8.3 Application Impact

This issue may cause a system hang that would be recoverable only by a Power On Reset (POR) or Watchdog reset.

4.8.4 Workarounds

To avoid accessing memories that are in a powered down, unclocked or error state, as per the above table, software may use a Memory Protection Unit (MPU) region to block explicit memory accesses, and then use a Data Synchronization Barrier (DSB) instruction and an Instruction Synchronization Barrier (ISB) instruction to ensure that all in-flight transactions have been completed. Any subsequent access to this region will trigger an MPU exception that can be handled in software. Protection violations stay in the CPU and generate exceptions. So, exceptions are raised as long as the MPU is configured to block all accesses.

If the memory to be powered down includes cacheable read/write data, this might result in cache-line evictions and the MPU region strategy described above is not able to block these accesses. Possible workarounds for this case are as follows:

- Do not power down MSPI/PSRAM devices if that memory is used for writeable data and thus might have dirty cache lines in the cached data.
- Clean the data cache before powering down MSPI, then implement a memory barrier, then use the MPU to block accesses. Note that there is a significant cost in cleaning the cache in terms of time and power.
- Make the PSRAM region write-through, then use the DSB/ISB and MPU strategy above to block further accesses. Note that this option has power and performance implications.

4.8.5 Erratum Resolution Status

There are no plans at this time to fix this erratum on the Apollo510 or Apollo510B SoC.

4.8.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not implement a workaround to prevent the described conditions that may cause a hang.

4.9 ERR010: IOS: FIFO read gets stuck/stalled

4.9.1 Description

If the SPI main pauses the SPI SCK, the Apollo510 may get stuck waiting for the next SCK from the SPI main 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.

4.9.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 SoC.

4.9.3 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).

4.9.4 Workarounds

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

4.9.5 Erratum Resolution Status

There are no plans to implement any further fix for this erratum in the Apollo510 family of SoCs.

4.9.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

4.10 ERR011: IOS: MISO line is not tri-stated when CE is de-asserted

4.10.1 Description

When configured as a SPI subordinate using the IOS module, the Apollo510 does not tri-state the MISO pin when CE is de-asserted. Instead, the MISO pin is driven static low when CE is driven high. This condition exists for the half-duplex as well as the full-duplex IOS instances.

4.10.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.10.3 Application Impact

If there are multiple subordinates on the same SPI bus as an Apollo510 configured as a SPI subordinate, other subordinate devices will be prevented from driving data onto the MISO line. In effect this prevents the sharing of the SPI bus with other subordinate devices.

4.10.4 Workarounds

Workarounds include:

1. Do not have any other subordinate devices on the SPI bus.
2. Add an external tri-state buffer between the Apollo510 MISO pin and the MISO line of the SPI bus so that the MISO pin on Apollo510 does not drive the bus when CE is de-asserted by the SPI main device.
3. Implement a software workaround that reconfigures and tri-states the Apollo510 MISO pin when the Apollo510 CE signal is de-asserted (driven high) by the SPI main device.

4.10.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.10.6 AmbiqSuite Workaround Status

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

4.11 ERR012: IOS: FIFO mode failures with certain FIFO_BASE and FIFOPTR settings

4.11.1 Description

There are two conditions which may cause FIFO mode to fail because of a conflict with the host-side 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.

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 host-side register space will cause a wraparound of the LRAM address which is not correct.

4.11.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 SoC.

4.11.3 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.

4.11.4 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 host-side register space.

4.11.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.11.6 AmbiqSuite Workaround Status

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

4.12 ERR013: IOS: Subordinate interrupt (SLINT) is not bonded out to external pin

4.12.1 Description

The half-duplex IOS0 Subordinate module's interrupt (SLINT) is a function select for GPIO162. However, this GPIO pad is not bonded out to an external pin on the Apollo510 BGA. Therefore, the IOS0's supported and built-in interrupt mechanism is not operational.

4.12.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 SoC.

4.12.3 Application Impact

The effect of this issue is that it requires either another GPIO to be used as the IOS0 interrupt pin, or the full-duplex IOS1/IOS2 combination to be used in place of the half-duplex IOS0.

4.12.4 Workarounds

Since the IOS Subordinate interrupt for the full-duplex IOS (SLFDINT) is operational, a workaround for this issue is to use that IOSFD module, either in half-duplex or the IOS1/IOS2 pair in full duplex mode. Another workaround for this issue on the Apollo510 SoC is to use another GPIO interrupt to mimic the SLINT behavior.

4.12.5 Erratum Resolution Status

There are no plans to fix this erratum on any existing or planned Apollo510 package.

4.12.6 AmbiqSuite Workaround Status

The second workaround stated above is implemented in the AmbiqSuite SDK. In the half-duplex IOS examples, ios_fifo and ios_burst, the SLINT_GPIO is set to 1 to use another GPIO interrupt to mimic the SLINT behavior for the IOS0 in the Apollo510 SoC.

4.13 ERR014: MEMORY: Restrictions on use of DMA to TCM

4.13.1 Description

There are three known restrictions on DMAing to TCM.

1. A DMA transfer cannot cross a 4 kB boundary.
2. Access to TCM when it is powered down returns an incorrect error condition (Success).
3. Access to TCM while the CPU is in deepsleep with memory retained returns an error.(1)

4.13.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.13.3 Application Impact

This issue may cause multiple issues including data loss/corruption or bus hang when DMAing to TCM.

4.13.4 Workarounds

A workaround for the 4 kB boundary issue is to ensure that DMA transfers are constructed in software so as not to cross a 4 kB boundary in TCM.

There are no workarounds for the other two issues; the application must ensure that TCM is not powered down when an access is attempted and that access to TCM must not occur when the CPU is in deepsleep.

4.13.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 or /Apollo510B devices. This erratum is intended to be fixed in future derivatives of the Apollo5 family.

4.13.6 AmbiqSuite Workaround Status

There is no workaround in the AmbiqSuite SDK for this issue. The user should avoid this issue in the application.

1. This is correct behavior of the Arm® M55 architecture. It is mentioned here to notify the user of this TCM usage restriction.

4.14 ERR015: MEMORY: SIMO Buck cannot be enabled via INFO0 setting

4.14.1 Description

The SIMO buck cannot be enabled automatically via INFO0.

4.14.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

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 currently are no plans 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 ERR016: MSPI: Peripheral accesses not guaranteed when clock source <= 48 MHz

4.15.1 Description

When an MSPI clock source lower than or equal to 48 MHz is selected, not all MSPI peripheral accesses can be guaranteed. Thus, for MSPI0 and MSPI3, only HFRC_96MHz, HRFC2_125MHz, HFRC_192MHz or HRFC2_250MHz may be selected in the CLKGEN_MSPIIOCLKCTRL_MSPInIOCLKSEL register fields. For MSPI1 and MSPI2, only HFRC_96MHz or HFRC_192MHz may be selected.

4.15.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs.

4.15.3 Application Impact

With clock options lower than 96 MHz disabled, the lower limit of the MSPI's SCLK at a supported divider of 32 (set in the MSP_DEV0CFG_CLKDIV0) is increased.

4.15.4 Workarounds

There is no workaround available for this erratum.

4.15.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.15.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK has limited the MSPI clock source selection to be higher than 48 MHz in the HAL.

4.16 ERR018: MSPI: Crossing a memory boundary during DMA transfer causes device hang

4.16.1 Description

A DMA transfer which has a start address near the upper boundary of a memory block and the transfer extends into the next memory block, a device hang occurs. The root cause is that a DMA burst is fixed to 0, so the DMA address will not change when it crosses the memory boundary.

This condition applies for several boundaries, which are the DTCM - SRAM0, SRAM0 - SRAM1 and SRAM1 - SRAM2 boundaries. The address ranges for the affected memory spaces are the following for the Apollo510:

- DTCM: 0x20000000 - 0x2007FFFF
- SSRAM0: 0x20080000 - 0x2017FFFF
- SSRAM1: 0x20180000 - 0x2027FFFF
- SSRAM2: 0x20280000 - 0x2037FFFF

In addition, this susceptibility to device hang occurs when a DMA transfer crosses a 4 kB boundary in DTCM.

4.16.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of Apollo510 and Apollo510B SoCs.

4.16.3 Application Impact

This issue causes a device hang and requires a reset to recover.

4.16.4 Workarounds

A workaround for this issue is to ensure that a DMA transfer does not attempt to cross the DTCM - SSRAM memory address boundary.

4.16.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.16.6 AmbiqSuite Workaround Status

It is up to the user to ensure that a memory boundary crossing in DMA transfers does not occur.

4.17 ERR019: MSPI: Upper data lines are pulled low instead of staying in high impedance mode

4.17.1 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.

4.17.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.17.3 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.

4.17.4 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.

4.17.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.17.6 AmbiqSuite Workaround Status

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

4.18 ERR020: MSPI: Mixed Mode 1-1-4 does not work as expected

4.18.1 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. Read operation is not affected and therefore it does not pose a problem for XIP/XIPMM reads.

4.18.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.18.3 Application Impact

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

4.18.4 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.

4.18.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.18.6 AmbiqSuite Workaround Status

The workaround code is released in all three NAND flash drivers of the AmbiqSuite SDK.

4.19 ERR021: PDM: High OSR causes aliasing and PDM mic distortion to occur during the PDM-to-PCM conversion

4.19.1 Description

In sampling configurations/frequencies with high Oversampling Ratio (> 128), the audio from the digital mic has distortion during speech. Most of the distortion comes from aliasing which is occurring during the PDM-to-PCM conversion. With a DMIC clock of 3.072 MHz and an OSR of 192, there is distortion from 1.6 kHz to 16 kHz. When OSR=192, the MIC input signal bandwidth is greater than Fs/2 (= 16/2 = 8 kHz). If the mic input signal spectrum is wider than 8 kHz, a significant amount of aliasing occurs. Therefore, a maximum OSR of 128 is supported.

4.19.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B.

4.19.3 Application Impact

This issue may cause poor audio quality in certain sampling configurations.

4.19.4 Workarounds

A workaround for this issue is to not use a sampling configuration with an OSR of > 128. The use of either of the following configurations is recommended:

- Fpdm_cko = 3.072 MHz, Fs = 24 kHz, OSR=128
- Fpdm_cko = 1.536 MHz, Fs = 16 kHz, OSR=96

4.19.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 or Apollo510B devices. This erratum is intended to be fixed in future derivatives of the Apollo5 family.

4.19.6 AmbiqSuite Workaround Status

There is no workaround implemented in the AmbiqSuite SDK. It is up to the user to configure the PDM with a sampling setting which does not result in unacceptable aliasing.

4.20 ERR022: PWRCTRL: APB bus hang when accessing unpowered OTP registers or INFO space

4.20.1 Description

An access to the OTP info space or OTP registers when the OTP is powered down will result in an APB bus hang. The OTP power domain is powered up by setting the PWRCTRL_DEVPWREN_PWRENOTP field to Enable. This field is enabled by default.

4.20.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.20.3 Application Impact

This issue may cause an APB bus hang which would require a power-on reset to recover.

4.20.4 Workarounds

There is no workaround for this issue. The application must ensure OTP is powered on before attempting to access OTP registers or OTP infospace. Please note that while OTP is powered on at reset, for low power operation it should be turned off whenever not in use.

4.20.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 or Apollo510B devices. This erratum is intended to be fixed in future derivatives of the Apollo5 family.

4.20.6 AmbiqSuite Workaround Status

There is no workaround implemented in the AmbiqSuite SDK for this issue. The OTP is powered on at reset and should be turned off whenever not in use, especially for low power operation. The application must power on the OTP before attempting to access OTP registers or OTP infospace.

4.21 ERR023: PWRCTRL: CPU cannot go into deepsleep when either OTP or ROM is powered on

4.21.1 Description

The SoC design requires that the CPU be powered on when either OTP or ROM is powered on.

4.21.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.21.3 Application Impact

Without proper handling in software, the OTP and ROM power domains will see large leakage current when the CPU is powered down. Please note that for low power operation, the recommendation is to turn off OTP and ROM whenever not in use, even if CPU is active.

4.21.4 Workarounds

The workaround for this issue is to make sure that OTP and ROM are off when CPU is inactive (in deepsleep).

4.21.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 or /Apollo510B devices. This erratum is intended to be fixed in future derivatives of the Apollo5 family.

4.21.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK prevents the CPU from going into deepsleep if either OTP or ROM is still powered on.

4.22 ERR024: PWRCTRL: MCU continues to use the clock for LP mode (HFRC) after switching from LP to HP mode

4.22.1 Description

After transitioning from low power (LP) mode to high performance (HP) mode, the MCU still uses the clock for LP (HFRC) after the switch if the clock for HP (HFRC2) is not enabled before switching. For the HP HFRC2 clock to immediately go into effect upon transition from LP mode to HP mode, the HFRC2 clock must be enabled and allowed a settling time for the HFRC2 clock to stabilize before the transition.

4.22.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.22.3 Application Impact

The MCU still being clocked by the LP mode clock, HFRC, after transitioning from LP mode to HP mode may be unexpected, and the slower than expected clock rate may need to be accounted for.

4.22.4 Workarounds

A workaround for this issue is to enable HFRC2 and allow a wait time of up to 15 µs for the HFRC2 clock to stabilize before switching to HP mode.

4.22.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.22.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK has integrated the software workaround described above into the PWRCTRL HAL. When waking up from deepsleep to HP mode, the CPU wakes up even if not operating in HP mode yet. During the transition period, the CPU runs in LP mode (clocked by HFRC). The AM_HAL_STALL_CPU_HPWAKE in the PWRCTRL HAL should be set to 1 if it is necessary to block the CPU after waking up until running in HP mode (clocked by HFRC2). Note that the default setting of 0 for AM_HAL_STALL_CPU_HPWAKE is better for reducing wakeup latency.

4.23 ERR025: PWRCTRL: SIMO Buck fails to transition from Active Mode to LP Mode

4.23.1 Description

An automatic SIMO Buck transition mode, which is enabled by setting both PWRCTRL_TONCNTRCTRL register fields LPMODESWOVR and ENABLELPOVR to 0x1, causes the SIMO Buck to transition to active mode if there is an increase in load current, but otherwise stays in Low Power (LP) Mode. Once the SIMO Buck transitions into Active Mode in deepsleep mode, it periodically retries to go into LP Mode. The SIMO Buck can transition from LP to Active if the temperature change and load change are gradual. However, it cannot handle the transition if there is a rapid change in temperature or load current. Because of this limitation, production trims do not make use of SIMO Buck automatic mode transitioning, and it is recommended that these PWRCTRL bits remain cleared.

4.23.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.23.3 Application Impact

In automatic SIMO Buck transition mode, if the simobuck transitions into Active Mode in deepsleep mode and fails to transition back into Low Power (LP) Mode, there could be a high deepsleep current (over 2 mA).

4.23.4 Workarounds

The workaround is to not use the SIMO Buck automatic transition mode. Instead, before entering deepsleep mode monitor the temperature and force the SIMO Buck into active mode if the temperature is higher than 50 degrees C and put/leave the SIMO Buck in LP mode if lower than 50 degrees C. After entering deepsleep, or even if in active mode, perform a periodic check of the temperature to see if the SIMO Buck needs to or can transition its power mode.

4.23.5 Erratum Resolution Status

There are no plans at this time to fix this erratum since the current production power trims do not require SIMO Buck automatic mode transitioning.

4.23.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK’s Spot Manager HAL helps the application manage SOC power states across temperature.

4.24 ERR026: PWRCTRL: Some power domains might not power up as expected after a SWPOI reset

4.24.1 Description

All the power domains can be trimmed to be powered on after a SWPOI reset. In the failure case, some of the power domains might not power up as expected after a SWPOI reset. This is a hardware startup issue which could result in non-functional peripherals/memories.

4.24.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B.

4.24.3 Application Impact

This issue may cause some power domains to not be powered up as expected, making it not functional and resulting in some or all memories having content different than that before the SWPOI reset.

4.24.4 Workarounds

A workaround for this issue is to check the power enable/status fields to ensure that each one's power enable bit matches its power status bit. This is done by performing register reads, comparing values and taking needed action after any reset for differences in any of the following register fields in the PWRCTRL set of registers:

1. SSRAMPWREN with SSRAMPWRST
2. DEVPWREN with DEVPWRSTATUS
3. MEMPWREN with MEMPWRSTATUS (except PWRSTCACHE)
4. AUDSSPWREN with AUDSSPWRSTATUS

If there is a mismatch of any power enable field and corresponding power status field (e.g., PWREN=1 and PWRST=0), perform the following steps:

1. Power off the specific domain (set PWREN=0).
2. Wait for 2.5 µs.
3. Power on (set PWREN=1).
4. Wait for PWRST to become set (should not take more than 50 µs).

4.24.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 or /Apollo510B devices. This erratum is intended to be fixed in future derivatives of the Apollo5 family.

4.24.6 AmbiqSuite Workaround Status

The workaround for this issue is implemented in the pre-installed SBL and hence is not visible to or a part of a user application.

4.25 ERR027: RSTGEN: Brown-out reset status bit may get set inadvertently during power-up

4.25.1 Description

The brown-out reset status bit, RSTGEN_STAT_BORSTAT, may inadvertently get set due to the brown-out signals powering up inconsistently during system power-up. Whether the status bit gets set is influenced by the voltage levels of the power supplies (VDDH/VDDP/VDDA) and the ramp rate of these supplies.

4.25.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.25.3 Application Impact

This issue may cause the brown-out reset status bit to get set incorrectly during system power-up.

4.25.4 Workarounds

The recommendation is to clear this status bit (RSTGEN_STAT_BORSTAT) after initial power-up. The RSTGEN_STAT register is a mirror of the actual hardware status register (RSTSTAT), which has already been cleared by bootrom after any reset event including initial power up. The bootrom and/or the secure bootloader maintain and update the RSTGEN_STAT register with appropriate status settings after each reset event. While this register is intended to be read-only, all bits in the register, including reserved bits, are in fact writable. Therefore, application software should take great care when writing this register.

4.25.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.25.6 AmbiqSuite Workaround Status

An SDK HAL workaround is implemented in the function am_hal_pwrctrl_low_power_init(). If that function is not called by the application, the recommendation is to implement one of the following workarounds:

1. Execute the following code: if (RSTGEN->STAT_b.POASTAT) { RSTGEN->STAT_b.BORSTAT = 0; }
2. On a production line, after initial power up, a debugger writes the field to 0.
3. Any reset event after power up will clear the BORSTAT status bit and resolve the issue.

4.26 ERR028: RTC: CB field value is unpredictable when year = 99 and CEB = 1

4.26.1 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 randomly, which is (essentially) every RTC clock.

4.26.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.26.3 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 from 1999 to 2000.

4.26.4 Workarounds

The workaround for this issue is to keep the CEB cleared, disabling the CB bit from automatically toggling when the CTRYR register field rolls over from 99 to 00. The century value should be maintained in the CB field by software.

4.26.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.26.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK has been modified as follows:

- CEB is no longer supported. The bit should be kept cleared (default) when setting up the RTC in the HAL.
- No automatic toggle of the CB field when the CTRYR field rolls over from 99 to 00.

Note that the Century Bit, RTC_CTRUP_CB, which is used to specify the century (for leap year calculation purposes), was not previously documented correctly in the Apollo510 register set and has been corrected. This bit should be set to 1 if the century is 20xx, and it should be set to 0 if the century is 21xx.

4.27 ERR029: STIMER: Constraints on writing to SCMPRn registers and handling Compare interrupts

4.27.1 Description

Compare interrupts are delayed by one STIMER clock. Additionally, 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 4 STIMER clock cycles.

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

4.27.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.27.3 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.

4.27.4 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 STIMER clock cycles. For Apollo510 and Apollo510B SoCss, this latency correction is 3 cycles. Therefore, the minimum valid delta setting is 4 STIMER clock cycles.

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.

4.27.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.27.6 AmbiqSuite Workaround Status

The AmbiqSuite HAL contains the workaround for these delays 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, the HAL accounts 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.

4.28 ERR030: SYSPLL: Special case where EXTREFCLK cannot be set as the SYSPLL clock

4.28.1 Description

For Apollo510 SoC:

The SYSPLL can be clocked by either the high-speed crystal oscillator (XTALHS) or an external reference clock (EXTREFCLK) through a glitch-less mux within the SYSPLL. If XTALHS had been enabled at any time after reset and then disabled, then the selection of the EXTREFCLK as the SYSPLL clock source cannot be made. If the XTALHS had been enabled and then disabled, it must be re-enabled to be able to transition the SYSPLL clock source from XTALHS (default setting) to the EXTREFCLK.

If XTALHS had not been enabled prior to selecting the clock source for the SYSPLL or is not present in the system, then EXTREFCLK can be selected as the SYSPLL clock source directly (by setting the MCUCTRL_PLLCTL0_FREFSEL bit to EXTREFCLK).

For Apollo510B SoC:
The 48 MHz crystal, BLE XTAL, connected directly to the BLE Controller module on the Apollo510B, is also divided by 4 and becomes the EXTREFCLK reference clock. This 12 MHz clock serves as the (only) reference clock for the SYSPLL. Even though the XTALHS crystal pins are not present on the Apollo510B, an internal connection still exists from the XTALHS internal oscillator to the mux which controls the clock input to the SYSPLL, and in fact is selected by default as the SYSPLL reference clock via the MCUCTRL_PLLCTRL0_FREFSEL register field. Just as is the case with the Apollo510, if XTALHS had been enabled at any time after reset and then disabled, then the selection and connection of the 12 MHz EXTREFCLK as the SYSPLL clock source cannot be made. The (disabled) XTALHS must first be selected as the SYSPLL reference clock (default case) and not have been enabled and disabled before switching SYSPLL's reference clock to EXTREFCLK. Note that the SYSPLL itself should not be enabled until its reference clock selection is switched from the default selection (XTALHS) to EXTREFSEL.

4.28.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.28.3 Application Impact

This issue may prevent the SYSPLL from selecting the desired clock source or either clock source.

4.28.4 Workarounds

For Apollo510 SoC:

A software workaround for this issue is to have both clocks available and enabled when configuring the SYSPLL to use the EXTREFCLK if the XTALHS had been enabled and disabled prior to SYSPLL clock selection. The EXTALHS can be disabled after SYSPLL clock selection if it is not needed elsewhere.

Note that if XTALHS had not been enabled prior to selecting the clock source for the SYSPLL or is not present in the system, then EXTREFCLK can be selected as the SYSPLL clock source directly (by setting the MCUCTRL_PLLCTL0_FREFSEL bit to EXTREFCLK).

For Apollo510B SoC:

A software workaround for this issue is to ensure that the following sequence is followed in selecting the reference clock for the SYSPLL:

1. Leave the XTALHS disabled (default setting).
2. Ensure that the BLE XTAL has started up completely and the BLE XTAL oscillator output clock is stable.
3. Select the EXTREFCLK as the clock source for the SYSPLL (MCUCTRL_PLLCTRL0_FREFSEL = EXTREFCLK).
4. Enable the SYSPLL (MCUCTRL_PLLCTRL0_SYSPLLPDB = ENABLE). Note that there may be other steps in preparation of enabling the SYSPLL.

4.28.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 or Apollo510B SoC. This erratum is intended to be fixed in future derivatives of the Apollo5 family.

4.28.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK implements the software workaround for this issue and has implemented the Clock Manager HAL, which includes managing the input clock for the SYSPLL.

4.29 ERR031: USB: Induced D+ output pulse may cause unintended disconnect

4.29.1 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 a disconnect one, causing the host USB SW to report about the unexpected disconnect from Apollo510. 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.

4.29.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.29.3 Application Impact

The Apollo510 may fail to connect or stay connected with USB hosts that deviate from strict USB 2.0 specifications.

4.29.4 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 Apollo510 and Apollo510B SoCs Datasheet.

4.29.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.29.6 AmbiqSuite Workaround Status

No workaround is needed in the AmbiqSuite SDK.

4.30 ERR032: DEBUG: SWO pin goes low during deepsleep

4.30.1 Description

The enabled SWO pin goes low during deepsleep mode which violates the SWO specifications. This signal is expected to stay high during deepsleep. The SWO output is in the PDDBG power domain which does remain powered when the M55 core goes to sleep. However, these signals are routed through the MCUH power domain which is powered down in deepsleep resulting in the SWO signal being isolated low.

4.30.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.30.3 Application Impact

This issue causes loss of the use of the SWO output when the device enters deepsleep mode.

4.30.4 Workarounds

A workaround for this issue is to ensure the SWO print pipeline is flushed and that the SWO pin is set high and configured as a GPIO before going into deepsleep. When exiting deepsleep, the GPIO should be reconfigured for the SWO function.

4.30.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 or Apollo510B SoC. This erratum is intended to be fixed in future derivatives of the Apollo5 family.

4.30.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides a function, am_bsp_debug_printf_deepsleep_prepare(), as a reference for users to work around this issue. The user must ensure all printing has completed before calling this function, and it should be called just before and just after going to deepsleep. Please note that implementing this workaround impacts deepsleep power, as the system will only reach the CORE_DEEPSLEEP level rather than the SYS_DEEPSLEEP level. Also see the PWRCTRL_SYSPWRSTATUS register.

4.31 ERR033: Boot Loader: Keybank restrictions when used for OTA encryption

4.31.1 Description

Keybank1, Keybank2 and Keybank3 keys cannot be used as AES keys for use with secure OTAs or wired updates. Keybank0 keys (4 total at kek index 0x80-0x83) can be used.

4.31.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 and Apollo510B SoCs.

4.31.3 Application Impact

If keys in a keybank other than Keybank0 are used for AES encryption, this issue causes encryption errors when used in conjunction with SBL based OTAs.

There is no impact on usage of hardware keys KCP and KCE (key index 0 and 1). They can continue to be used. Also, usage of the keybanks outside of Ambiq SBL is not restricted.

4.31.4 Workarounds

Use only Keybank0 keys (kek index 0x80-0x83). They can be used as AES keys for secure OTAs or wired updates through Ambiq SBL.

4.31.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 or Apollo510B SoC. This erratum is intended to be fixed in future derivatives of the Apollo5 family.

4.31.6 AmbiqSuite Workaround Status

No workaround is needed in the AmbiqSuite SDK.

4.32 ERR034: Boot Loader: SBL's OEM recovery does not use INFO0 settings for wired update

4.32.1 Description

The Secure Boot Loader (SBL) does not use the settings for wired recovery from INFO0, but instead uses the default settings in INFO1 even when INFO0 is valid and the settings are configured. The flag used by the SBL to determine which settings to use, INFO0 or INFO1, is not initialized by the time the OEM recovery is initiated, so the SBL always picks up the INFO1 settings.

4.32.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs.

4.32.3 Application Impact

This issue causes the SoC to always use factory settings in INFO1 for the SBL regardless of the setting of the INFO0/INFO1 selection flag. This issue comes into play only when doing an application and GPIO-initiated recovery. It works as intended, i.e., INFO0 settings will be used, for OEM recovery in case the OEM certifications fail.

4.32.4 Workarounds

There is no workaround for this issue.

4.32.5 Erratum Resolution Status

This issue is fixed in SBL version 1.16 or later and can be applied to all Apollo510/Apollo510B devices.

4.32.6 AmbiqSuite Workaround Status

This issue is fixed in the SBL for versions 1.16 or later, and is in AmbiqSuite SDK releases 5.0.0 and later. SDK5.0.0 includes an SBL OTA update image that can be used to update earlier devices.

4.33 ERR036: Secure Boot ROM: SDIO1 can only be used in 1-bit mode for MRAM recovery

4.33.1 Description

SDIO1 can only be used in 1-bit mode for MRAM recovery. SDIO0 is fully functional. The boot ROM recovers the SBL (~64 KB), and the SBL recovers the OEM's image (up to 256 KB).

4.33.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs.

4.33.3 Application Impact

This issue requires that SDIO0 be used for MRAM recovery when 4-bit or 8-bit mode is desired.

4.33.4 Workarounds

A workaround for this issue is to use SDIO0 which allows the SBL to work in 1, 4 or 8-bit mode.

4.33.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 or Apollo510B SoC.

4.33.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not affect, and is not affected by, this issue.

4.34 ERR037: IOS/IOSFD: Erroneous XCMPRF interrupt triggered

4.34.1 Description

When using the IOS or IOSFD, the XCMPRF (Read from FIFO space transfer complete) flag is incorrectly set, instead of the XCMPRR (Read from register transfer complete) flag, on host reads of the 0x7C FIFOCNTLO (FIFO Counter Low-Byte) Host Address Register. This same behavior is seen during host reads of 0x7D FIFOCNTUP (FIFO Counter High-Byte) Host Address Register) as well.

A read by the host of 0x7C (or 0x7D) triggers the XCMPRF interrupt rather than the XCMPRR interrupt. A host read of FIFO data at 0x7F also triggers the XCMPRF interrupt as would be expected. Therefore, depending on the host reads, extra interrupts during read operations could occur.

4.34.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs.

4.34.3 Application Impact

This issue will cause extraneous and misleading interrupts if XCMPRF interrupts are enabled.

4.34.4 Workarounds

Possible workarounds for this issue are the following:

1. Do not use 0x7C/7D commands to get the FIFO counter. Instead use the direct access area to store the data size and exchange it during the handshake.
2. Add a FIFO space check in the interrupt routine to check for the amount of space used in the FIFO to filter the extra interrupt and make sure the read operation is complete.

4.34.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.34.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK examples use the first workaround to handle this issue.

4.35 ERR038: BootROM: Clock for MSPI used for MRAM recovery not selectable

4.35.1 Description

When using an MSPI instance for MRAM recovery, the MSPI_READ_CLKSEL and MSPI_PRECMD_CLKSEL fields in the INFO0_NV_OPTIONS register are not processed (i.e., ignored) which means that the MSPI clock source (MSPInIOCLKSEL field of the CLKGEN_MSPIIOCLKCTRL register) always selects the default clock source for the MSPI module being used. The INFO0 function of selecting the clock source for the MSPI used for MRAM recovery has been deprecated for this series of SoCs, and the default clock for the MSPI will always be used.

The default clock for all MSPIs, as well as the clock divider selected by the INFO0_NV_CONFIG0 register which gets copied to the CLKDIV0 field of the MSPI_DEV0CFG register, work as expected.

4.35.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs.

4.35.3 Application Impact

If an MSPI instance is to be used as the MRAM recovery channel, this issue restricts the input clock source for the chosen MSPI to its default clock. The final MSPI clock speed is the resulting frequency after clock divide as determined by the setting of the CLKDIV0 field of the MSPI_DEV0CFG register.

4.35.4 Workarounds

The workaround for this issue is to use the clock divider in the INFO0_NV_CONFIG0 register to achieve an appropriate clock frequency for the memory/storage device being used.

4.35.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 SoC or Apollo510B SoC.

4.35.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK and the BootROM provide all necessary functionality to program the INFO0 registers for MRAM recovery.

4.36 ERR039: DC: Incorrect pixel data output order from the DC SPI

4.36.1 Description

Incorrect pixel data output order from the DC SPI causes color distortion in the displayed image. This is a reported NemaDC hardware erratum: Not supported: 24-bit/16-bit/8-bit color formats with layer size which is not a multiple of 4.

4.36.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs.

4.36.3 Application Impact

This issue may cause a distorted image on the display.

4.36.4 Workarounds

A workaround for this issue is to set the framebuffer resolution to a multiple of 4x4, slightly larger than the panel resolution.

4.36.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 SoC or Apollo510B SoC.

4.36.6 AmbiqSuite Workaround Status

No change is needed in the AmbiqSuite SDK.

4.37 ERR040: DEBUG: Cannot connect to a debugger after DEBUG domain is powered down

4.37.1 Description

After SWO printing has concluded, the Apollo510 SDK HAL functions disable the Instrumentation Trace Macrocell (ITM) and Trace Port Interface Unit (TPIU), then power down the DEBUG domain. Before doing so, the HAL functions attempt to detect that all printing has completed via ITM status signals. Based on these signals, in some cases it's possible that the disabling and DEBUG domain power down occur before the ITM/TPIU pipeline has actually become quiescent. When this happens, a debugger (e.g. J-Link Commander) will not be able to connect again to the device as the DEBUG domain will not power up until a power cycle occurs.

4.37.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs.

4.37.3 Application Impact

This issue could preclude the use of a debugger after SWO printing and subsequent ITM/TPIU disabling and power down of the DEBUG domain without a power-cycle of the device.

4.37.4 Workarounds

A workaround for this issue is to insert a 500 µs delay, even after all ITM statuses indicate that SWO printing has completed. If there is an attempt to power down the DEBUG domain immediately after SWO printing, there may still be pending activity in the SWO path or the ITM/TPIU pipeline which can prevent the DEBUG domain from shutting down properly. This is because the hardware state machine is still busy or waiting for the last bits of trace data to be flushed. Inserting a 500 µs wait after ITM statuses and before disabling/powering-down provides time to ensure that all SWO/ITM/TPIU activity has completed and that the pipeline is flushed.

4.37.5 Erratum Resolution Status

There are no plans at this time to fix the erratum in a future silicon revision of the Apollo510 or Apollo510B SoCs.

4.37.6 AmbiqSuite Workaround Status

An SDK HAL function, am_hal_itm_tpiu_pipeline_flush(), is provided that includes the 500 µs delay and ensures that all printing and flushing has completed.

4.38 ERR041: DEBUG: DEBUG power domain may not power down under certain conditions

4.38.1 Description

The DEBUG power domain can be put into a state where it will not power down after reads from the processor ROM table, which leads to increased power consumption and manifests as the PWRCTRL_DEVPWRSTATUS_PWRSTDBG register field getting stuck at 1. Due to a handshake issue in the power control logic, this can happen when accessing the processor ROM table with back-to-back reads. This issue occurs with the following conditions and effect:

1. The issue occurs whether or not the debug domain already had been powered up before the back-to-back reads occur.
2. Only ROM table reads are at risk of causing this issue.
3. Once in this state, the PWRSTDBG cannot be cleared by software and a chip reset is required.

4.38.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs.

4.38.3 Application Impact

This issue causes increased power consumption by forcing the debug domain to remain powered up during normal operation.

4.38.4 Workarounds

A workaround that prevents the issue is to insert a short delay between back-to-back reads of the processor ROM table / JEDEC registers. Ambiq recommends that the delay be at least 1 µs.

4.38.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 SoC or Apollo510B SoC.

4.38.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK Apollo510 HAL functions that access the ROM table have been updated in SDK5.1.0 to include a 1 µs delay between reads.

4.39 ERR042: I2S: Error flag is not set when a DMA error occurs

4.39.1 Description

The I2S does not set an error flag, TXDMASTAT_TXDMAERR or RXDMASTAT_RXDMAERR, or generate an error interrupt when a DMA error occurs. This error condition could cause the DMACPL logic to not work properly, and it may occur even when the DMACPL bit gets set.

4.39.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs

4.39.3 Application Impact

An undetected DMA error may cause anomalies (stop or glitch) in the audio.

4.39.4 Workarounds

The application software should add a timeout when waiting for the TXDMASTAT_TXDMACPL or RXDMASTAT_RXDMACPL flag to get set in case a DMA error occurs preventing the DMA Complete flag from asserting. If a timeout occurs while waiting for the TXDMACPL or RXDMACPL flag to get set, clear (write a 0 to) the TXDMASTAT_TXDMAERR or RXDMASTAT_RXDMAERR bit, respectively, regardless of its read value, which clears the internal error logic. In addition, since a DMA error could have occurred even when the DMACPL bit got set, the error could cause the DMACPL logic to not work properly. Therefore, the DMAERR bit should be cleared (1) upon each setting of the DMACPL bit (because an error could still have occurred), and (2) upon timeout for the DMACPL bit not getting set.

4.39.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 SoC or Apollo510B SoC.

4.39.6 AmbiqSuite Workaround Status

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

4.40 ERR043: MRAM: Concurrent AXI burst read to MRAM and an MRAM write cause read data to be corrupted

4.40.1 Description

A write to INFO0, INFO1 or MRAM main_mem coincident with an AXI burst read results in read data corruption.

4.40.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs

4.40.3 Application Impact

This issue causes incorrect data to be read from INFO spaces or MRAM main_mem.

4.40.4 Workarounds

A workaround for this issue is as follows.

During an NVM / MRAM write, a simultaneous read to MRAM might, in rare cases, result in the corruption of the read data returned to the requester. The contents of the MRAM are not affected, so subsequent reads would return the correct data. The managers that could be affected are:

1. M55 - Data being written to MRAM should not be read from MRAM.
2. GPU - Graphics objects read from MRAM by the GPU may in rare cases be corrupted. This may result in this graphics asset being corrupted for this one read instance. A subsequent read will return the correct data (if no write is in progress). Customer can decide if this is acceptable or if MRAM should be avoided by the GPU.
3. APBDMA – Similarly, if any device is using the DMA function to read from MRAM during an MRAM write, then in rare cases, the read data could be corrupted. We recommend not to use MRAM as the source of DMA data to avoid this issue.
4. DC – The Display Controller burst size is not compatible with MRAM. The DC should never read from MRAM.

4.40.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 SoC or Apollo510B SoC.

4.40.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not restrict when storage or memory is accessed. It is up to the application software to avoid concurrent reads and writes that may cause read data being corrupted, such as described in this erratum.

4.41 ERR044: MRAM: Powered down NVM1 causes state machine to hang

4.41.1 Description

A bug exists where, if NVM1 is powered down while NVM0 is powered on, the MRAM Controller state machine hangs upon a POR or POI level reset.

4.41.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs that have two MRAM banks.

4.41.3 Application Impact

If the MRAM state machine is hung because NVM1 was powered down upon a POR/POI level reset, another POR/POI level reset after powering down NVM0, or a POA level interrupt (without the need to power down NVM0), would be required to reset the state machine.

If a POA level reset occurs when NVM1 was powered down, the state machine does not get hung so no subsequent (forced) reset is needed.

4.41.4 Workarounds

To prevent this issue from occurring, ensure one of the following any time a POR/POI level reset may occur:

1. NVM1 is powered on
2. Both NVM0 and NVM1 are powered off
3. NVM0 and NVM1 are not in different power states

In order to recover from a situation where NVM1 was disabled when a device reset occurred and the MRAM state machine is hung, power off NMV0 (clear the PWRCTRL_MEMPWREN_PWRENNVM bit) to make both NVM0 and NVM1 in the off state before doing a POR/POI level reset. This must be executed in a location other than NVM such as ITCM or SSRAM. A POA level reset will also be able to reset the state machine without the need to power down NVM0 from a non-NVM location.

4.41.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 SoC or Apollo510B SoC.

4.41.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK cannot prevent the occurrence of NVM1 being powered down when a pin reset occurs. It is the developer's responsibility to ensure that conditions which cause the MRAM state machine to hang do not occur.

4.42 ERR045: TMR: Timer does not start counting until 4th clock tick

4.42.1 Description

Since two timer clock cycles are required for the TMREN to be synchronized from the APB clock to the timer clock and another two timer clock cycles are required to synchronize the initial value, a total of four extra timer clock cycles must occur before the timer counter starts to increment, regardless of the clock selected for the timer. Subsequent clocking is not affected.

4.42.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 and Apollo510B SoCs

4.42.3 Application Impact

This issue causes a 4 timer-clock-cycle delay before the timer starts to count. if this delay is not taken into consideration, then the timing of the first clock output will not be as expected.

4.42.4 Workarounds

A workaround for this issue is to compensate for the four additional timer clock cycles required for the first timer cycle to complete. This can be done by setting the TIMER_TMRnCMP0 register to a value that is 4 lower for the first timer “half” cycle (i.e., time to transition) than what would be set in a normally-timed “half” cycle. Note that this workaround only works for timer “half” cycles that are longer than 4 timer cycles.

4.42.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 SoC or Apollo510B SoC.

4.42.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not provide an automatic workaround for this erratum. It is the responsibility of the developer to implement the above or another suitable workaround for this issue.

4.43 ERR046: USB: High leakage current in USB PHY

4.43.1 Description

There is a high-current leakage path drawing about 34 mA from the 3.3 V VDDUSB33 rail when the digital PHY rail (0.9 V) is not powered.

4.43.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 SoC and Apollo510B SoC.

4.43.3 Application Impact

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

4.43.4 Workarounds

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

4.43.5 Erratum Resolution Status

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

4.43.6 AmbiqSuite Workaround Status

The workaround for this issue is a hardware change that does not require any software update.

5. Ordering Information

Table 3: Ordering Information for Apollo510 SoC and Apollo510B SoC

Device Name	Commercial Temp Range (-20°C to 70°C)	Industrial Temp Range (-40°C to 85°C)	Package Type	GPIOs	NVM (MRAM)[a]	SRAM	Connectivity	Package[b] Size (mm)
Apollo510 SoC	AP510NFA-CBR	AP510NFA-IBR	BGA	183	4 MB	3.75 MB	No Connectivity	6.6 x 6.6 x 0.75 225-pin BGA
Apollo510 SoC	AP510NFA-CCR	AP510NFA-ICR	WLCSP	144	4 MB	3.75 MB	No Connectivity	4.9 x 4.7 182-pin WLCSP
Apollo510B SoC	AP510BFA-CBR	-	BGA	96	4 MB	3.75 MB	Bluetooth Low Energy	5.6 x 5.6 x 0.8 153-pin WFBGA

a. Factory-installed Cortex-M4 firmware reduces the amount of available NVM which varies for each series derivative. The amount of NVM used will be made available in a future datasheet release.
b. Packing: Tape and Reel

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

SE-A510-3p0
Version 3.0
December 2025