ERRATA LIST

Apollo510 Lite Series SoCs

Ultra-low Power Apollo SoC Family

Doc. ID: SE-A510LS-1p0

Doc. Revision: 1.0, July 2026

Legal Information and Disclaimers

AMBIQ MICRO BELIEVES THE INFORMATION IN THIS DOCUMENT IS ACCURATE AT THE TIME OF PUBLICATION. HOWEVER, THE INFORMATION MAY CONTAIN TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. AMBIQ MICRO RESERVES THE RIGHT TO MAKE CORRECTIONS, MODIFICATIONS, ENHANCEMENTS, IMPROVEMENTS, OR OTHER CHANGES TO ITS PRODUCTS (AND DOCUMENTATION) AT ANY TIME WITHOUT NOTICE.

INFORMATION IN THIS DOCUMENT IS PROVIDED SOLELY TO ENABLE THE USE OF AMBIQ MICRO PRODUCTS AND IS PROVIDED “AS IS.” NO LICENSE IS GRANTED TO DESIGN OR FABRICATE ANY INTEGRATED CIRCUITS BASED ON THE INFORMATION IN THIS DOCUMENT. THIS DOCUMENT MAY NOT BE USED IN CONNECTION WITH ANY LEGAL ANALYSIS CONCERNING THE AMBIQ MICRO PRODUCTS DESCRIBED HEREIN. AMBIQ MICRO DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT.

AMBIQ MICRO PRODUCTS ARE SOLD SUBJECT TO AMBIQ MICRO’S TERMS AND CONDITIONS OF SALE; SOFTWARE IS PROVIDED PURSUANT TO APPLICABLE LICENSE TERMS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, IS GRANTED UNDER ANY INTELLECTUAL PROPERTY RIGHT OF AMBIQ MICRO OR ANY THIRD PARTY BY THIS DOCUMENT.

CUSTOMERS ARE SOLELY RESPONSIBLE FOR THE DESIGN, VALIDATION, TESTING, AND SECURITY OF THEIR PRODUCTS AND APPLICATIONS, INCLUDING THE SPECIFIC MANNER IN WHICH AMBIQ MICROS PRODUCTS ARE INCORPORATED. “TYPICAL” PARAMETERS ARE FOR REFERENCE ONLY AND IT IS THE CUSTOMER’S RESPONSIBILITY TO VALIDATE “TYPICAL” SPECIFICATIONS FOR THEIR APPLICATION.

AMBIQ MICRO PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN APPLICATIONS WHERE PRODUCT FAILURE COULD RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE (“HIGH-RISK APPLICATIONS”) WITHOUT EXPRESS WRITTEN APPROVAL. IF PRODUCTS ARE USED IN HIGH-RISK APPLICATIONS WITHOUT SUCH APPROVAL, THE CUSTOMER SHALL INDEMNIFY AND HOLD HARMLESS AMBIQ MICRO AND ITS AFFILIATES FROM ALL RESULTING CLAIMS, DAMAGES, COSTS, AND EXPENSES, INCLUDING REASONABLE ATTORNEYS’ FEES. IN NO EVENT SHALL AMBIQ MICRO BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, OR PUNITIVE DAMAGES, INCLUDING LOST PROFITS OR DATA, ARISING FROM THE USE OF THIS DOCUMENT OR ANY PRODUCT.

“AMBIQ”, “APOLLO”, “SPOT”, AND COMBINATIONS THEREOF ARE TRADEMARKS OF AMBIQ MICRO, INC. OTHER PRODUCT NAMES USED IN THIS PUBLICATION ARE FOR IDENTIFICATION PURPOSES ONLY AND MAY BE TRADEMARKS OF THEIR RESPECTIVE COMPANIES.

Table of Content

Silicon Errata for the **Apollo510 Lite Series SoCs**

1. Introduction

This document is a detailed compilation of known device errata for the general availability revisions of the Apollo510 Lite Series SoCs. Unless stated otherwise, all listed errata apply to all SoCs in the series and all packages of the SoCs.

2. Document Revision History

Table 1: Document Revision History

Rev No.	Date	Description
1.0	Jul 2026	Initial Release

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	Affected Silicon Revisions	Resolution Status	Workaround
“ERR010: IOS: FIFO read gets stuck/ stalled” on page 8	All existing	No fix planned	No workaround
“ERR012: IOS: FIFO mode failures with certain FIFO_BASE and FIFOPTR settings” on page 9	All existing	No fix planned	Software workaround
“ERR014: MEMORY: Restrictions on use of DMA to TCM” on page 10	All existing	No fix planned	Software workaround
“ERR016: MSPI: Peripheral accesses not guaranteed when clock source <= 48 MHz” on page 11	All existing	No fix planned	No workaround
“ERR018: MSPI: Crossing a memory boundary during DMA transfer causes device hang” on page 12	All existing	No fix planned	Software workaround
“ERR019: MSPI: Upper data lines are pulled low instead of staying in high impedance mode” on page 13	All existing	No fix planned	No workaround
“ERR020: MSPI: Mixed Mode 1-1-4 does not work as expected” on page 14	All existing	No fix planned	Software workaround
“ERR027: RSTGEN: Brown-out reset status bit may get set inadvertently during power-up” on page 15	All existing	No fix planned	Software workaround
“ERR029: STIMER: Constraints on writing to SCMPRn registers and handling Compare interrupts” on page 16	All existing	No fix planned	Software workaround
“ERR030: SYSPLL: Special case where EXTREFCLK cannot be set as the SYSPLL clock” on page 18	All existing	No fix planned	Software workaround
“ERR031: USB: Induced D+ output pulse may cause unintended disconnect” on page 19	All existing	No fix planned	No workaround
“ERR032: DEBUG: SWO pin goes low during deepsleep” on page 20	All existing	No fix planned	Software workaround
“ERR037: IOS/IOSFD: Erroneous XCMPRF interrupt triggered” on page 21	All existing	No fix planned	Software workaround

Table 2: Errata Summary

Erratum Number, Title and Page	Affected Silicon Revisions	Resolution Status	Workaround
“ERR038: BootROM: Clock for MSPI used for MRAM recovery not selectable” on page 22	All existing	No fix planned	Software workaround
“ERR039: DC: Incorrect pixel data output order from the DC SPI” on page 24	All existing	No fix planned	Software workaround
“ERR042: I2S: Error flag is not set when a DMA error occurs” on page 25	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 26	All existing	No fix planned	Software workaround
“ERR045: TMR: Timer does not start counting until 4th clock tick” on page 27	All existing	No fix planned	Software workaround
“ERR046: USB: High leakage current in USB PHY” on page 28	All existing	No fix planned	Hardware workaround
“ERR047: GPIO: Failure to generate GPIO interrupt in dual edge mode” on page 29	All existing	No fix planned	Hardware and software workaround
“ERR049: MRAM: GPU-initiated NVM burst reads limited to 32 bytes” on page 30	All existing	No fix planned	Software workaround
“ERR051: PDM: Clock status bit does not work” on page 31	All existing	No fix planned	Software workaround
“ERR052: PDM: 24-bit data stored in 32-bit word without sign extension” on page 32	All existing	No fix planned	Software workaround
“ERR054: RTC: Failure to read the latest counter value” on page 33	All existing	No fix planned	Software workaround
“ERR055: BootROM: MSPI2 not usable for MRAM recovery” on page 34	All existing	No fix planned	Hardware and 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 ERR010: IOS: FIFO read gets stuck/stalled

4.1.1 Description

If the SPI main pauses the SPI SCK, the Apollo510 Lite 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.

4.1.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.1.3 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).

4.1.4 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.

4.1.5 Erratum Resolution Status

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

4.1.6 AmbiqSuite Workaround Status

There is no software workaround in the AmbiqSuite SDK.

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

4.2.1 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.

4.2.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.2.3 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.

4.2.4 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.

4.2.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.2.6 AmbiqSuite Workaround Status

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

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

4.3.1 Description

A DMA transfer cannot cross a 4 kB boundary.

4.3.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 Lite Series SoCs.

4.3.3 Application Impact

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

4.3.4 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.

4.3.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.3.6 AmbiqSuite Workaround Status

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

4.4 ERR016: MSPI: Peripheral accesses not guaranteed when clock source <= 48 MHz

4.4.1 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 all MSPI instances, only HFRC_96MHz, HFRC_192MHz, PLLVCO_125MHz or PLLVCO_250MHz may be selected in the CLKGEN_MSPIIOCLKCTRL_MSPInIOCLKSEL register fields. Note that the *upper* specified limit of an MSPI instance's I/O clock (after source divider) is dependent on the particular MSPI instance being used and the mode of operation, i.e., Standard 4-Wire Serial / SDR Non-DQS or DDR with DQS. Consult the MSPI electrical specifications of the latest datasheet for these upper limits.

4.4.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.4.3 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.

4.4.4 Workarounds

There is no workaround available for this erratum.

4.4.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.4.6 AmbiqSuite Workaround Status

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

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

4.5.1 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 to two boundaries - the DTCM-SRAM0 and SRAM0-SRAM1 boundaries. The address ranges for the affected memory spaces are the following for the Apollo510 Lite:

- DTCM: 0x20000000 - 0x2007FFFF
- SSRAM0: 0x20080000 - 0x2017FFFF
- SSRAM1: 0x20180000 - 0x2023FFFF

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

4.5.2 Affected Silicon Revisions

This silicon erratum applies to all existing revisions of Apollo510 Lite Series SoCs.

4.5.3 Application Impact

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

4.5.4 Workarounds

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

4.5.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.5.6 AmbiqSuite Workaround Status

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

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

4.6.1 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.

4.6.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 Lite Series SoCs.

4.6.3 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.

4.6.4 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.

4.6.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.6.6 AmbiqSuite Workaround Status

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

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

4.7.1 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.

4.7.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 Lite Series SoCs.

4.7.3 Application Impact

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

4.7.4 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.

4.7.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.7.6 AmbiqSuite Workaround Status

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

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

4.8.1 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.

4.8.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 Lite Series SoCs.

4.8.3 Application Impact

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

4.8.4 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 Boot ROM after any reset event including initial power up. The Boot ROM 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.

4.8.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.8.6 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.

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

4.9.1 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 three 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.

4.9.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 Lite Series SoCs.

4.9.3 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.

4.9.4 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 Lite Series SoCs, 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.

4.9.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.9.6 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.

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

4.10.1 Description

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).

4.10.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 Lite Series SoCs.

4.10.3 Application Impact

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

4.10.4 Workarounds

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).

4.10.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.10.6 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.

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

4.11.1 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 Lite. 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.

4.11.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 Lite Series SoCs.

4.11.3 Application Impact

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

4.11.4 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 Lite Series SoCs Datasheet.

4.11.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.11.6 AmbiqSuite Workaround Status

No workaround is needed in the AmbiqSuite SDK.

4.12 ERR032: DEBUG: SWO pin goes low during deepsleep

4.12.1 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.

4.12.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of Apollo510 Lite Series SoCs.

4.12.3 Application Impact

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

4.12.4 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.

4.12.5 Erratum Resolution Status

There are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.12.6 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.

4.13 ERR037: IOS/IOSFD: Erroneous XCMPRF interrupt triggered

4.13.1 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.

4.13.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.13.3 Application Impact

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

4.13.4 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.

4.13.5 Erratum Resolution Status

There currently are no plans to fix this erratum.

4.13.6 AmbiqSuite Workaround Status

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

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

4.14.1 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.

If MSPI0, whose default clock is the system PLL (CLKGEN_MSPIIOCLKCTRL_MSPI0IOCLKSEL = PLLVCL_250MHz), is selected as the MRAM recovery channel, the PLL does not get initialized and therefore the MSPI channel does not work in BootROM.

The clock divider selected by the INFO0_NV_CONFIG0 register, which gets copied to the CLKDIV0 field of the MSPI_DEV0CFG register, works as expected for all MSPI default clocks.

Note that INFO1 sets the default PLL on and configures it for 25 MHz output. It runs during the SBR, and then the SBL turns it off if there is no MRAM recovery in progress, or once it is done otherwise. So MSPI0 can be used for MRAM recovery, with the PLL being its default clock source with a frequency of 25 MHz.

4.14.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.14.3 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. For MSPI0, which uses the System PLL as its default clock source, cannot be used (without workaround) since the PLL does not get initialized in the BootROM.

4.14.4 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.

Also, a workaround for the MSPI0 issue where the System PLL does not get initialize in the BootROM is to turn on the System PLL by default by changing the INFO1 trim settings. This causes a small increase in power consumed during bootup and therefore the SBL will be updated to turn off System PLL as soon as possible.

4.14.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.14.6 AmbiqSuite Workaround Status

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

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

4.15.1 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.

4.15.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.15.3 Application Impact

This issue may cause a distorted image on the display.

4.15.4 Workarounds

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

4.15.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.15.6 AmbiqSuite Workaround Status

No change is needed in the AmbiqSuite SDK.

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

4.16.1 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.

4.16.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.16.3 Application Impact

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

4.16.4 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.

4.16.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.16.6 AmbiqSuite Workaround Status

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

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

4.17.1 Description

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

4.17.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.17.3 Application Impact

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

4.17.4 Workarounds

A workaround for this issue is as follows.

During an NVM / MRAM write, a simultaneous dual-beat read (32 bytes) to MRAM might, in rare cases, result in the corruption of the last 16 bytes of 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. DC – The Display Controller burst size is not compatible with MRAM. The DC should never read from MRAM.

Note that reads from CM4 and the DMA are safe because they only generate single-beat reads (16 bytes).

4.17.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.17.6 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.

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

4.18.1 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.

4.18.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.18.3 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.

4.18.4 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.

4.18.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.18.6 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.

4.19 ERR046: USB: High leakage current in USB PHY

4.19.1 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.

4.19.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.19.3 Application Impact

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

4.19.4 Workarounds

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

4.19.5 Erratum Resolution Status

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

4.19.6 AmbiqSuite Workaround Status

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

4.20 ERR047: GPIO: Failure to generate GPIO interrupt in dual edge mode

4.20.1 Description

GPIOs fail to generate a GPIO interrupt in dual-edge mode (GPIO_PINCFGn_IRPTENn = 3).

4.20.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.20.3 Application Impact

Dual-edge GPIO interrupt mode is not supported on this device.

4.20.4 Workarounds

The workaround to achieve dual-edge mode operation is by tying two GPIOs together, with one using rising edge interrupts and the other using falling edge interrupts.

4.20.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.20.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK enables the software part of the above workaround.

4.21 ERR049: MRAM: GPU-initiated NVM burst reads limited to 32 bytes

4.21.1 Description

The NVM/MRAM only supports bursts up to 32 bytes in length. Longer bursts return invalid data for the portion above 32 bytes.

4.21.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.21.3 Application Impact

This issue may cause incorrect data being read from NVM via a GPU-initiated burst longer than 32 bytes.

4.21.4 Workarounds

A workaround for this issue is to not use a burst longer than 32 bytes or ignore any read data after the first 32 bytes.

4.21.5 Erratum Resolution Status

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

4.21.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK does not restrict the size of a burst. It is the responsibility of the user application to handle this issue according to the stated workaround.

4.22 ERR051: PDM: Clock status bit does not work

4.22.1 Description

The read-only APBPDMCLKACTIVE bit never asserts to indicate that the clock is ready in the CRM_APBPDMCRM register even though the PDM clock has been enabled.

4.22.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.22.3 Application Impact

Since the APBPDMCLKACTIVE bit never asserts even though the PDM clock has been enabled, the firmware cannot poll this bit for clock status.

4.22.4 Workarounds

After powering up the PDM module and enabling the clock, the software should add a delay of 1 µs before attempting to access the PDM module registers, instead of polling the APBPDMCLKACTIVE bit to check the clock status.

4.22.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.22.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK provides the ability to set a delay but does not automatically implement a delay before accessing PDM registers as a workaround for this issue

4.23 ERR052: PDM: 24-bit data stored in 32-bit word without sign extension

4.23.1 Description

The 24-bit PDM data is stored as a 32-bit word (int32_t) in the FIFOREAD register or the DMA buffer, with the upper 8 bits unused (always 0). For example, a -2 PCM sample is stored as 0x00FFFFFE, but the CPU interprets this as a 32-bit value of 16,777,214 instead of -2.

4.23.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.23.3 Application Impact

This issue may cause a misinterpretation of the read PDM data if additional processing is not performed in software.

4.23.4 Workarounds

A workaround for this issue is to restore the sign bit by having the software left-shift the value by 8 bits and then right-shift it by 8 bits when reading the FIFOREAD field or the DMA buffer. This operation introduces additional processing overhead.

4.23.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.23.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK performs the above workaround to prevent misinterpretation of PDM data.

4.24 ERR054: RTC: Failure to read the latest counter value

4.24.1 Description

A single read-back of the Cortex-M55 RTC counter register fails to read the latest counter value and instead returns the last value read.

4.24.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.24.3 Application Impact

When the RTC counter value is read, it will appear as if the RTC counter is not incrementing, thereby resulting in incorrect RTC time keeping with the application firmware reporting an incorrect real-time measurement.

4.24.4 Workarounds

Use a triple read when reading the RTC counter register as follows: If first 2 RTC counter back-to-back reads return the same value, use first value, otherwise read again and use the value of the 3rd read.

4.24.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.24.6 AmbiqSuite Workaround Status

The AmbiqSuite SDK executes the triple-read described in Workarounds in am_hal_rtc_time_get() and the associated value compare.

4.25 ERR055: BootROM: MSPI2 not usable for MRAM recovery

4.25.1 Description

Due to the MSPI2's D8 data line not being pinned out on the CSP package and the BootROM not supporting the CLKOND4 option (CLKOND4 INFO0 configuration setting), MSPI2 cannot be used for MRAM recovery in a CSP package, and the CLKOND4 option can not be used on any MSPI peripheral in either the BGA or CSP package.

4.25.2 Affected Silicon Revisions

This silicon erratum applies to all revisions of the Apollo510 Lite Series SoCs.

4.25.3 Application Impact

This issue precludes the use of MSPI2 for MRAM recovery. For that capability, another interface channel/ MSPI must be connected and used.

4.25.4 Workarounds

A workaround for this issue is to use another MSPI instance other than MSPI2.

4.25.5 Erratum Resolution Status

There currently are no plans to fix this erratum on any future silicon revision of the Apollo510 Lite Series SoCs.

4.25.6 AmbiqSuite Workaround Status

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

5. Ordering Information

Table 3: Ordering Information for Apollo510 Lite Series SoCs

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 (SSRAM and TCM)	Connectivity	Package[b] Size (mm)
Apollo510 Lite	AP510NLA-CCR	-	CSP	68	2 MB	2 MB	No Connectivity	4.047 x 3.948 x 0.510(max) 110-pin
Apollo510 Lite	AP510NLA-CBR	-	BGA	120	2 MB	2 MB	No Connectivity	5.3 x 5.3 x 0.8(max) 169-pin
Apollo510B Lite	-	AP510BLA-ICR	CSP	68	2 MB	2 MB	Bluetooth Low Energy	4.047 x 3.948 x 0.510(max) 110-pin
Apollo510D Lite	AP510DLA-CCR	-	CSP	68	2 MB	2 MB	Dual-Mode Bluetooth	4.047 x 3.948 x 0.510(max) 110-pin
Apollo510D Lite	AP510DLA-CBR	AP510DLA-IBR	BGA	120	2 MB	2 MB	Dual-Mode Bluetooth	5.3 x 5.3 x 0.8(max) 169-pin

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-A510LS-1p0
Version 1.0
July 2026