MAX32670/MAX32671
ERRATA SHEET

Revision A1 Errata

The errata listed below describe situations where components of this revision perform differently than expected or differently than described in the data sheet. Analog Devices may, at its own discretion, take future steps to correct these errata when the opportunity to redesign the product presents itself. Prior to that, Analog Devices has determined the following potential workarounds that customers may want to consider when addressing one of the situations described below.

This errata sheet only applies to components of this revision. These components are branded on the topside of the package with a six-digit code in the form yywwRR, where yy and ww are two-digit numbers representing the year and work week of manufacture, respectively, and RR is the revision of the component. To obtain an errata sheet on other die revisions, visit the Maxim website at www.maximintegrated.com/errata.

1) DEVICE IS ONLY RESET BY POR WHEN FLASH MEMORY PAGES ARE LOCKED

Description:
The device is only reset by a POR if the flash memory protection is enabled. This can occur:

- When the bootloader executes the LOCK or PERMLOCK command.
- If the flash memory protection is enabled through a GDB command.

Workaround:
Only use POR to reset the device when one or more flash memory pages are locked. Do not use GCR_RSTR0.sys, the external RSTN pin, or the internal WDT timer to generate a reset.

2) LPUART STOP BIT ONE BIT TIME LONGER THAN EXPECTED

Description:
The length of the stop bit selected by the UARTn_CTRL.stopbits field is: (15238)

0: 2 stop bits

1: 2.5 stop bits (for 5 bit mode) or 3 stop bits (for 6/7/8 bit mode)

Workaround:
None.

3) 32kHz OSCILLATOR DOES NOT DISABLE IN LOW-POWER MODES

Description:
The 32kHz oscillator is always enabled when the device is in any mode other than ACTIVE, regardless of whether GCR_CLKCN.xclken has been set to 1 or 0. (15243)

Workaround:
None.

419-100067; Rev 2; 8/22

4) DEVICE MUST OPERATE FROM 100MHz DEFAULT OSCILLATOR

Description:
The device will not operate as expected if a system oscillator other than the default IPO is selected. (15280)

Workaround:
None. Do not change the system oscillator from its default value.

5) DEVICE PINS ASSOCIATED WITH LPTIMER AND LPUART INSTANCES WILL NOT FUNCTION AS GPIOS AFTER THE PERIPHERAL CLOCK IS ENABLED

Description:
All of the pins associated with an instance of an LPTIMER or LPUART will no longer operate with the GPIO functionality after the first time that peripheral clock is enabled. All of the alternate functions associated with the device pins operate correctly. (15276)

Workaround:
Do not use the device pins associated with an instance of an LPTIMER or LPUART once its peripheral clock has been enabled.

6) WDT FEED SEQUENCE NOT REQUIRED FOR ACCESS TO WDT_CTRL.en

Description:
The WDT_CTRL.en field is not write-protected by the WDT feed sequence. (15292)

Workaround:
None. On B version silicon, unique feed sequences will be required for resetting the WDT count, writing WDT_CTRL.en to 1, and writing WDT_CTRL.en to 0.

7) SOME GPIOS DO NOT ISOLATE WHEN V_{DD} = 0V

Description:
The maximum voltage on P0.[2, 4, 5, 7] must be equal to or less than V_{DD} + 0.3V.

Workaround:
None.

8) ERFO CANNOT BE THE SYSTEM CLOCK SOURCE WHEN ENTERING DEEPSLEEP

Description:
The ERFO is disabled when entering DEEPSLEEP even if GCR_PM.erfo_pd is cleared to 0. (15275)

Workaround:
Switch to a different oscillator before entering DEEPSLEEP.

9) SYSTEM CLOCK SOURCE CANNOT BE SWITCHED DIRECTLY FROM IBRO TO INRO

Description:
The device does not operate as expected when SYS_OSC is switched from IBRO to INRO. (15279)

Workaround:
Switch from IBRO to another clock source first, then switch from that clock source to INRO.

10) STANDARD DMA OPERATIONS INVOLVING ECC-ENABLED SYSTEM RAM INSTANCES MAY TRANSFER INCORRECT DATA

Description:
Incorrect data may be transferred if operations are performed on system RAM instances with their ECC enabled. (21206)

Workaround:
Use a secondary DMA channel in conjunction with a primary DMA channel to perform DMA operations to and from ECC-enabled memory.

Considerations:

- The secondary DMA channel must be enabled before the primary DMA channel.
- The value of DMA_CHn_CNT.cnt for the secondary channel must be:

  (4 x number_of_primary_bursts) + 1

  For example, if both DMA_CHn_CNT.cnt = 2048 and DMA_CHn_CTRL.burst_size = 32 for the primary channel, then 64 bursts are required. DMA_CHn_CNT.cnt for the secondary channel is therefore:

  (4 x 64) + 1 = 257

Use the following steps to configure both the secondary and primary DMA channels:

1. Configure a secondary DMA channel for a dummy transfer of data. The destination is an unused location in the memory map, so this use of the secondary channel does not affect any usable memory.

   a) Set DMA_CHn_SRC.addr to 0x4000 0000.
   b) Set DMA_CHn_DST.addr to 0x4000 0200.
   c) Clear DMA_CHn_CTRL.srcinc to 0 to disable the source automatic incrementing.
   d) Clear DMA_CHn_CTRL.dstinc to 0 to disable the destination automatic incrementing.
   e) Set DMA_CHn_CTRL.srcwd to 0b10.
   f) Set DMA_CHn_CTRL.dstwd to 0b10.
   g) Set DMA_CHn_CTRL.burst_size to 0b00011.
   h) Configure DMA_CHn_CTRL.pri of the secondary DMA channel to the same value as DMA_CHn_CTRL.pri of the primary DMA channel.
   i) Configure DMA_CHn_CNT.cnt to the value calculated above.

2. Configure the primary DMA channel based on the transfer width. The source and destination address of the primary DMA channel must be aligned to a 32-bit boundary.

   - For a transfer width of 4 bytes:
     a) DMA_CHn_CNT.cnt must be a multiple of 4.
     b) DMA_CHn_CTRL.burst_size must be a multiple of 4 bytes.

   - For a transfer width of 2 bytes:
     a) DMA_CHn_CNT.cnt must be a multiple of 8.
     b) DMA_CHn_CTRL.burst_size must be a multiple of 8 bytes.
     c) Set GCR_MEMCTRL.ramws_en to 1 to enable SRAM wait states.

   - For a transfer width of 1 byte:
     a) DMA_CHn_CNT.cnt must be a multiple of 4.
     b) DMA_CHn_CTRL.burst_size must be a multiple of 4 bytes.
     c) Set GCR_MEMCTRL.ramws_en to 1 to enable SRAM wait states.

11) GPIO0_INEN AND GPIO0_PS RESET VALUES ARE INCORRECT

Description:
For all reset events, the reset values are
- GPIO0_INEN = 0xFFFFFFFF
- GPIO0_PS = 0x00000000

Workaround:
None.

MAX32670/MAX32671
REV A1 ERRATA

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	6/21	Initial release	—
1	3/22	Added erratum 8	2
2	8/22	Removed erratum 5; added errata 8–11	2–4