MAX32650–MAX32652 ERRATA SHEET

Revision A4 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) I²S IN SLAVE MODE CAN RECORD INCORRECT DATA IF A PARTIAL WORD IS RECEIVED DURING LEFT-CHANNEL RECEPTION

Description:
While in slave mode, receipt of a partial/truncated word in the left data channel loads incorrect data into the I²S Rx FIFO.

Workaround:
1) Do not enable the I²S peripheral while an external I²S master is transmitting.
2) Ensure that an external I²S master begins all transmissions with a complete word.

2) DEVICE DOES NOT EXIT BACKGROUND MODE IF SYSTEM CLOCK FREQUENCY IS FASTER THAN THE LPCLK FREQUENCY

Description:
The device does not exit BACKGROUND mode if the system clock frequency is faster than the LPCLK frequency.

Workaround:
The software workaround for this erratum is implemented in the appropriate Analog Devices-supplied API.

3) HYP_CS1 SIGNAL IS NOT DRIVEN INACTIVE WHILE THE HYPERBUS IS IDLE

Description:
The output driver on the HYP_CS1 pin is disabled when the hyperbus is idle. Devices connected to the HYP_CS1 signal are not guaranteed to enter their deselected state.

Workaround:
Configure the GPIO associated with the HYP_CS1 pin to input mode with the internal strong/normal pull-up enabled. This pulls the HYP_CS1 signal to its inactive state while the hyperbus is idle.

419-100019; Rev 3; 5/22

4) SPI MODE 1 AND MODE 3 OPERATION MAY BE AFFECTED WHEN QSPIn_CLK_CFG.*SCALE* = 0

Description:
The following settings are invalid when operation the SPI is operating in mode 1 or mode 3: (13390)
SCALE = 0, CLOCKHI = 0, CLOCKLO = 0
SCALE = 0, CLOCKHI = 1, CLOCKLO = 1

Workaround:
Do not use the invalid settings. The operating speeds generated by all other field combinations are valid.

5) SPIXF BUS IDLE FEATURE CANNOT BE DISABLED

Description:
The SPIXF bus idle feature cannot be disabled. (14358)

Workaround:
Set the SPIXF_BUS_IDLE.busidle to 0x1 to approximate the operation of the SPIXF when the bus idle feature is disabled. This will have minimal effect on sequential code execution.

6) UART RECEIVER REQUIRES ONE EXTRA BIT TIME BETWEEN STOP AND START BITS

Description:
The UART requires an external transmitter to send at least one more stop bit than specified in the UART_CTRL0.stop field.

Workaround:
There are two workarounds:
1) Configure the transmitter to send at least one additional stop bit than specified in the UART_CTRL0.stop field.
2) Ensure the transmitter generates at least one idle bit time between byte transmissions.

7) DO NOT USE UART Tx AND Rx FUNCTIONALITY SIMULTANEOUSLY

Description:
The peripheral does not operate as expected if both the receive and transmit functions are used simultaneously.

Workaround:
None. Assign the transmit and receive functionality to different peripherals if full-duplex operation is required.

8) UART TRANSMITTER GENERATES SPURIOUS PULSE WHEN USING 7.3728MHz CLOCK SOURCE

Description:
The UART generates a negative pulse two PCLK periods wide on the UART_TX line one bit period before the falling edge of the start bit. (10689)

Workaround:
There are two workarounds:
1) Use PCLK as the clock source when transmitting.
2) Ensure the external receiver will reject the spurious pulse.

9) I²S/SPIMSS AND SPI0 SHARE THE SPI0 IRQ HANDLER

Description:
Interrupt sources from both the I²S/SPIMSS and SPI0 peripherals call the same SPI0_IRQHandler.

Workaround:
If using both I²S/SPIMSS and SPI0 simultaneously, software must interrogate both the I²S and SPI0 interrupt flags to determine which peripheral caused the interrupt. Both peripherals can generate interrupts.

10) I²S SIGNAL INCORRECTLY TRANSMITTED ON P0.22 WHEN I²S IS ENABLED AND SPI0 IS DISABLED

Description:
The I2S_LRCLK signal is incorrectly transmitted on P0.22 under the following conditions:

- P0.22 is configured as SPI0_SS0 (AF1).
- P2.5 is configured as I2S_LRCLK (AF1).
- I²S is enabled in master mode
- SPI0 is disabled.

Workaround:
To prevent the I2S_LRCLK signal from being transmitted on P0.22, configure P0.22 as GPIO.

MAX32650–MAX32652
REV A4 ERRATA

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	4/20	Initial release	—
1	12/20	Added errata 4 and 5	2
2	6/21	Added errata 6 and 7	2
3	5/22	Added errata 8, 9, and 10	2, 3