MAX32650–MAX32652
ERRATA SHEET

Revision A3 Errata

The errata listed below describe situations where components of this revision perform differently than expected or differently than described in the data sheet. Maxim Integrated Products, Inc. may, at its own discretion, take future steps to correct these errata when the opportunity to redesign the product presents itself. Prior to that, Maxim 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 has been implemented in the appropriate Maxim-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 pullup enabled. This pulls the HYP_CS1 signal to its inactive state while the hyperbus is idle.

419-100017; Rev 2; 6/21

4) DEVICE DOES NOT OPERATE AS EXPECTED IF CERTAIN GPIO PORT 2 PINS CHANGE STATE WHILE OPERATING FROM FLASH MEMORY

Description:
The device will execute incorrect instructions if the state of certain GPIO port 2 pins change while executing code from internal flash memory. The change of state can be caused by an external signal driven into the pin, or software changing the GPIO port registers associated with the affected pins. The affected pins are:
P2[7]
P2[8]
P2[14]
P2[15]
P2[16]
P2[24]
P2[28]

Workaround:
1) Do not use the affected pins and leave them unconnected while executing code from flash memory. Configure the GPIO with the strong pullup connected to the external supply. Do not change the state of the corresponding bits in the GPIO2_OUT register from their default value. Alternately the pin can be connected to a static external signal if that signal remains at V_{DDIO} while executing code from flash memory.
2) Copy the desired code into SRAM and ensure the device only executes code located SRAM by disabling interrupts. The affected port pins can be used as inputs or outputs without restrictions. Before resuming code execution from flash, ensure the signal on the device pin remains at a static V_{DDIO}.

5) 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.

6) 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.

7) 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.

8) 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.

MAX32650–MAX32652
REV A3 ERRATA

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	7/19	Initial release	—
1	12/20	Added errata 5 and 6	2
2	6/21	Added errata 7 and 8	3

Maxim Integrated 160 Rio Robles, San Jose, CA 95134 USA 1-408-601-1000