MAX32620
ERRATA SHEET

Revision C2 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., intends to correct these errata when the opportunity to redesign the product presents itself.

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 our website at ***www.maximintegrated.com/errata****.*

1) SPI MASTER MAY NOT OPERATE AS EXPECTED IF SDIO IS OUTPUT PRIOR TO SLAVE SELECT

Description:
Setting the SPIMn_SPCL_CTRL.ss_sample_mode field causes unexpected behavior.

Workaround:
Do not use the SPI master sample mode feature. This erratum affects all instances of the I^{2}C master peripheral.

2) SPI MASTER BIT-SIZED TRANSACTIONS DO NOT OPERATE IN FEEDBACK MODE

Description:
The SPIM does not operate as expected if all the following conditions are met:

The SPIM is configured for SCLK feedback mode.

The Size Units field in the SPI transaction header is set to 0 (bits).

The Size field is greater than 7.

Workaround:

Guarantee that the next header is not written to the FIFO until the Tx payload is complete (Tx FIFO word count = 0).

Alternately, a transaction of more than 7 bits can be decomposed into two transactions composed of 1 or more bytes and the final transaction of the remaining number of bits.

3) SPI SLAVE TRANSMIT FIFO WORD COUNT MAY BE INCORRECT IN SPI MODES 1/3

Description:
The Tx word count should be zero after all Tx data has been transmitted and the FIFO is empty. In Modes 1 and 3, if the Tx FIFO is empty and SCLK toggles while SS is inactive (master communicating to another slave), the register “Tx FIFO Entries” may read “1 byte.” Only the word counter value read by software is affected. The FIFO empty flag sets properly when the last byte was transmitted and remains set until the user clears it. The FIFO itself is unaffected and the following data is not effected.

Workaround:
Firmware work around is:

```
if (Tx No Data Interrupt == 0)
TX_COUNT_REMAINING = 0
else
    TX_COUNT_REMAINING = “Tx FIFO Entries Register”
```

4) SPI MASTER MAY TRANSMIT INCORRECT DATA IF Tx FIFO STALLS

Description:
An SPI master may transmit incorrect data under the following conditions:

1. The SPI master stalls (runs out of data in the middle of a transaction).

2. The SPI mode and SCK high or low settings do not match these combinations.

SPI MODE	SCK HIGH	SCK LOW
0	1	1–15
1	1–15	1
2	1–15	1
3	1	1–15

Workaround:

Use the SCK settings in the table shown above.

Slower baud rates are set through the SPIM clock scaling registers:

- CLKMAN_SYS_CLK_CTRL_11_SPI0.spi0_clk_scale
- CLKMAN_SYS_CLK_CTRL_12_SPI1.spi1_clk_scale
- CLKMAN_SYS_CLK_CTRL_13_SPI0.spi2_clk_scale

5) SPI SLAVE SELECT ACTIVITY MAY NOT GENERATE EXPECTED INTERRUPT

Description:
The option is provided to generate an interrupt whenever a master has asserted and/or deasserted the slave select signal on an enabled slave select pin.

Workaround:
Do not use the slave select toggle interrupt feature. Configure the slave select pin as a GPIO and use the GPIO interrupt function to detect when a master has asserted and/or deasserted the slave select pin.

6) I^{2}C PUSH/PULL MODE REGISTER IS WRITE ONLY

Description:
The IOMAN_I2CM[n]_REQ.push_pull_mode field is write only. Read-modify-write operations on the IOMAN_I2CM[n]_REQ register clear IOMAN_I2CM[n]_REQ.push_pull_mode. As a result, it is not possible to directly determine if the request has been accepted.

Workaround:
None.

7) IOMAN_I2CM[n]_ACK.PUSH_PULL_MODE FIELD DOES NOT MIRROR IOMAN_I2CM[n]_REQ. PUSH_PULL_MODE

Description:
The IOMAN_I2CM[n]_ACK.push_pull_mode field does not mirror the IOMAN_I2CM[n]_REQ.push_pull_mode field. As a result, it is not directly possible to verify if the corresponding pin has implemented push/pull feature. This affects all instances of the I^{2}C master peripheral.

Workaround:
None.

8) P1.6 DOES NOT OPERATE AS EXPECTED WHEN 32kHZ OUTPUT MODE ON P1.7 IS ENABLED

Description:
Enabling the 32kHz clock output mode on P1.7 configures P1.6 as an input with an internal pullup. The primary and secondary functions, pulse train output and timer input are unavailable in this configuration. P1.6 remains in this state until the 32kHz clock output mode on P1.7 is disabled. The internal pullup configuration increases power consumption in low power modes.

Workaround:
Do not use P1.6 except as a GPIO input while the 32kHz clock output mode is enabled.

9) GPIO_FREE_P1.[7:6] ARE INCORRECT WHEN 32kHZ OUTPUT MODE ON P1.7 IS ENABLED

Description:
When the 32kHz output mode on P1.7 is enabled, GPIO_FREE_P1.7 GPIO = 1 and GPIO_FREE_P1.6 = 0. Instead, the bits should be GPIO_FREE_P1.7 GPIO = 0 and GPIO_FREE_P1.6 = 1.

Workaround:
Software should recognize the bits are inverted when reading GPIO_FREE_P1.[7:6].

10) SPI MASTER SCLK DUTY CYCLE NOT 50% AT 48MHz AND V_{DDIO} ≈ 1.8V

Description:
When operating at 48MHz and V_{DDIO} = 1.8V, t_{MCH} ≈ 12ns and t_{MCL} ≈ 8ns. At V_{DDIO} = 3.3V, the duty cycle is approximately 50%. This may violate the minimum required setup time during slave read operations in modes 0, 2, and 3. This issue does not adversely affect operation in mode 1.

Workaround:
In modes 0 and 2, use the following procedure:

1. Set SPIMx_GEN_CTRL.enable_sck_fb_mode = 1.

2. SPIMx_GEN_CTRL.invert_sck_fb_clk = 1.

3. SPIMx_MSTR_CFG.inact_delay ≥ 1.

4. SPIMx_FIFO_CTRL.rx_fifo_af_lvl ≤ 29.

5. SPIMx_SPCL_CTRL.rx_fifo_margin = 3.

In mode 3, increase the low SCLK period by setting SPIMn_MSTR_CFG.sck_lo_clk = 2.

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	9/16	Initial release	—