Errata

SLG4653x

CE-GP-002

Abstract

This document contains the known errata for SLG46533, SLG46534, SLG46535, SLG46536, SLG46537, SLG46538 and the recommended workarounds.

1 Information

Packages	For SLG46533, SLG46537, SLG46538: 20-pin STQFN: 2 x 3 x 0.55 mm, 0.4 mm pitch 22-pin MSTQFN 2 x 2.2 x 0.55 mm, 0.4 mm pitch For SLG46534, SLG46535, SLG46536: 14-pin STQFN: 2 x 2.2 x 0.55 mm, 0.4 mm pitch

2 Errata Summary

Table 1: Errata Summary

Issue #	Issue Title
1	Counter Incorrect Operation after the Reset
2	FILTER cell does not filter out glitches
3	Incorrect I2C Reads of the 8-bit Counter Registers
4	ACMP additional IN- leakage current
5	I2C Access to the 16-Bit Counter Data Registers
6	GPO output latched from I2C interface activity without slave address or I2C write operation match when IO Latching is Enabled

3 Errata Details

3.1 Counter Incorrect Operation after the Reset

3.1.1 Effect

Counter

3.1.2 Conditions

After the Reset.

3.1.3 Technical Description

If the Counter Reset is asserted at the same time as a rising clock edge, it is possible that the Counter Data will be reset incorrectly and the counter output may appear faster than expected. This phenomenon appears more often as the clock frequency increases.

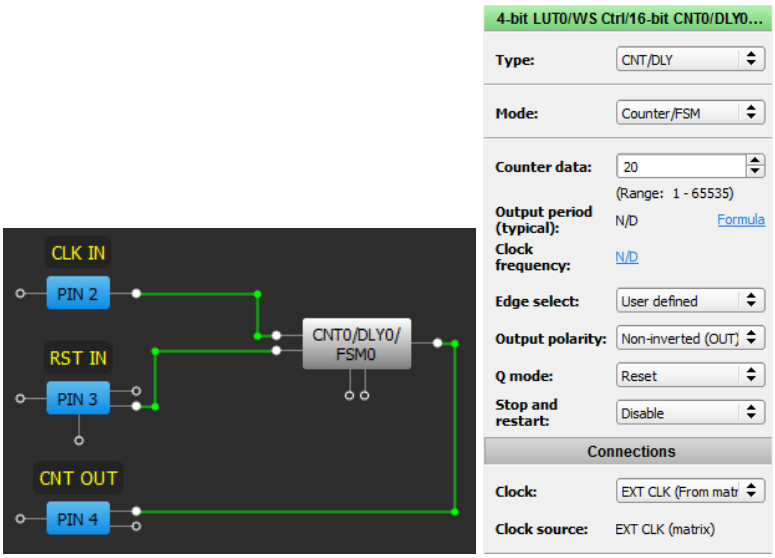

Figure 1

Figure 2

3.1.4 Workaround

Synchronize RESET input of the Counter with its CLK using 2 DFF cells as shown on Figure 3.

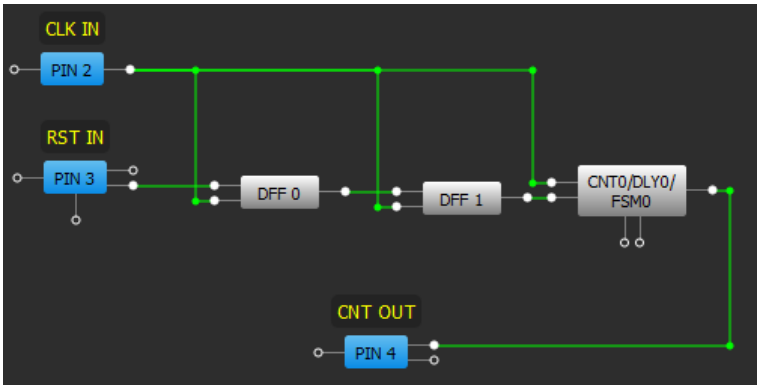

Figure 3

3.2 FILTER cell does not filter out glitches

3.2.1 Effect

FILTER

3.2.2 Conditions

When clock type is high frequency.

3.2.3 Technical Description:

If clock type high frequency input comes in, the FILTER cell may not filter out it. There are several factors like input frequency, duty cycle and LOW duration in such signal that may lead to its passing through FILTER block.

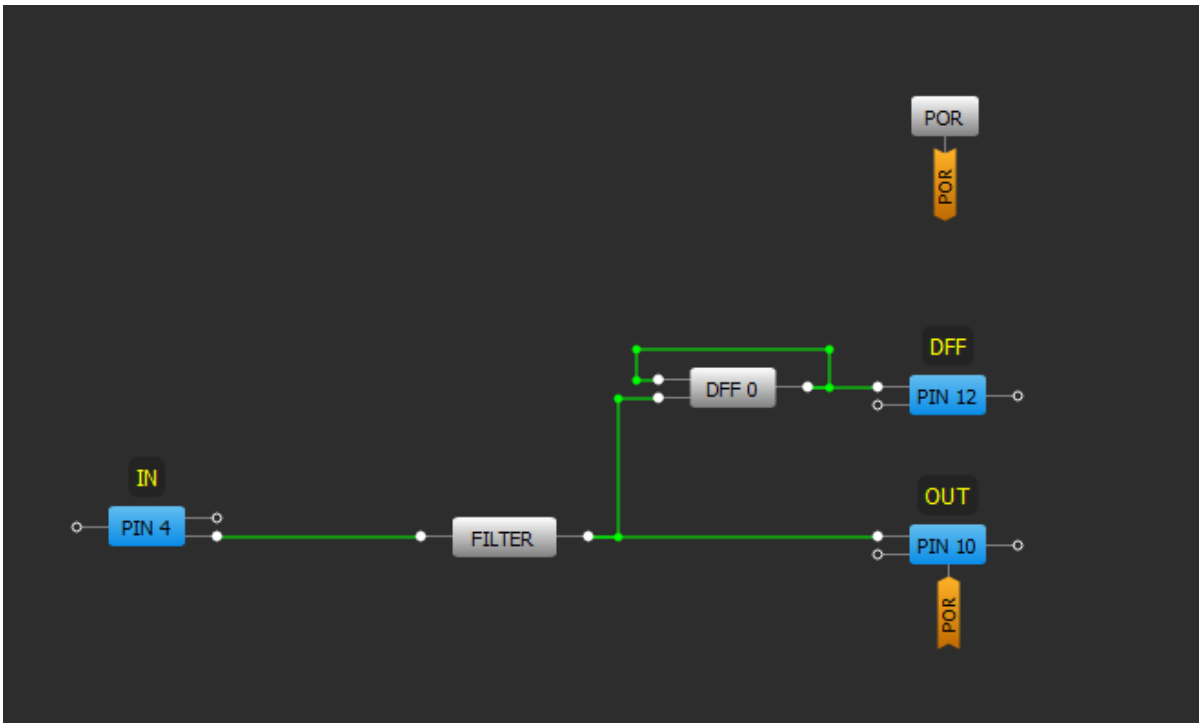

Figure 4

Channel 1 (yellow/top line) – PIN#4 (IN)

Channel 2 (light blue/2nd line) – PIN#10 (OUT)

Channel 3 (magenta /3rd line) – PIN#12 (DFF)

1. Period is 60ns. Pulse width is 10ns DC=16.7% (Correct functionality)

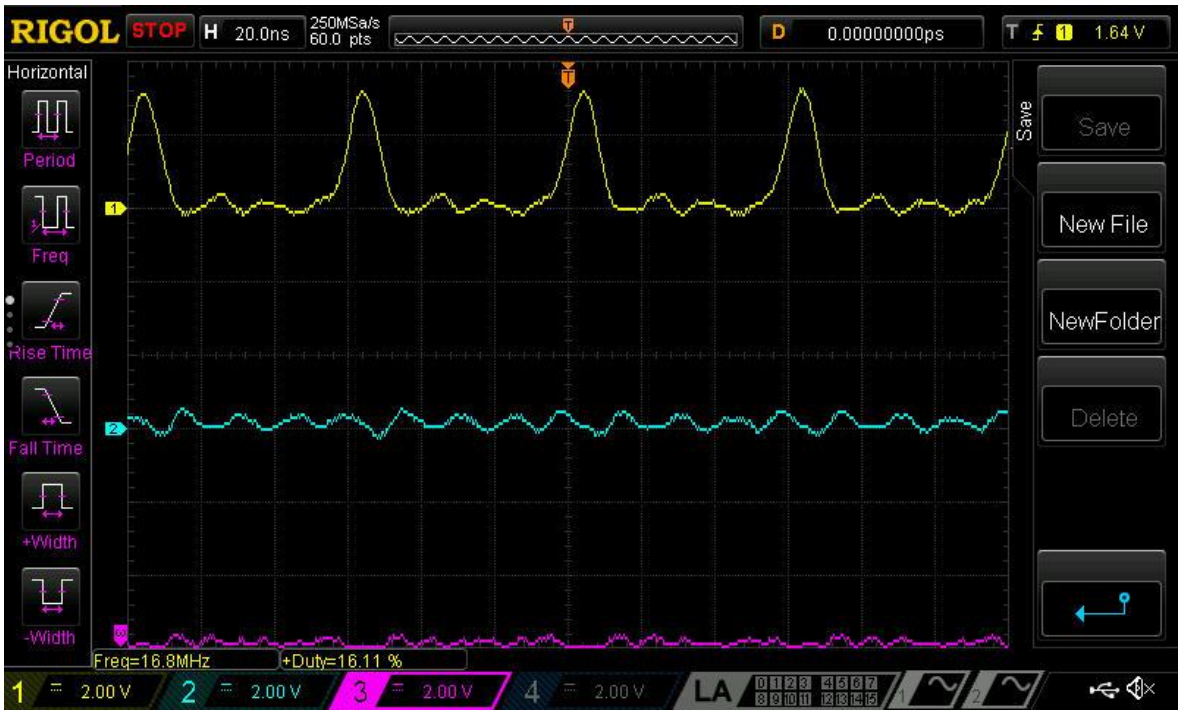

Figure 5

2. Period is 60ns. Pulse width is 20ns DC = 33.3% (Incorrect functionality)

Figure 6

3. Period is 60ns. Pulse width is 30ns DC=50% (Incorrect functionality)

Figure 7

4. Period is 60ns. Pulse width is 40ns DC=66.67% (Correct functionality)

Figure 8

3.2.4 Workaround:

Currently there is no workaround for this issue. Filter block is good at filtering short spontaneous glitches. It is intended to be used in series connection before the delay cell to avoid its latching.

3.3 Incorrect I2C Reads of the 8-bit Counter Registers

3.3.1 Effect

CNT2/DLY2 and CNT4/DLY4

3.3.2 Conditions

During asynchronous interaction between the CNT/DLY clock input and the I2C latch signal when the I2C latch signal and the clock input occur at about the same time.

3.3.3 Description:

Asynchronous interaction between the CNT/DLY clock input and the I2C latch signal (generated by an I2C read command of the CNT/DLY block's count value) can result in an incorrect I2C data read. The CNT/DLY block will count accurately, but the count value transferred into the block's I2C read register might be loaded incompletely if the I2C latch signal and the clock input occur at about the same time.

The example data capture below shows ten periodic I2C reads of CNT2/DLY2 configured to count down at about 16 clocks per read. The sixth read sample erroneously shows a value greater than that of the fifth. The seventh sample reads as if the previous I2C error never occurred - the difference from the fifth sample (176) to the seventh (143) is 33 clocks or 16 clocks + 17 clocks as expected.

Channel 1 (yellow/top line) – PIN#2 (CNT2/DLY2 Out)

Channel 2 (light blue/2nd line) – PIN#1 (I2C Read Triggers)

Channel 3 (magenta /3rd line) – PIN#8 (I2C SCL)

Channel 3 (dark blue /4th line) – PIN#9 (I2C SDA)

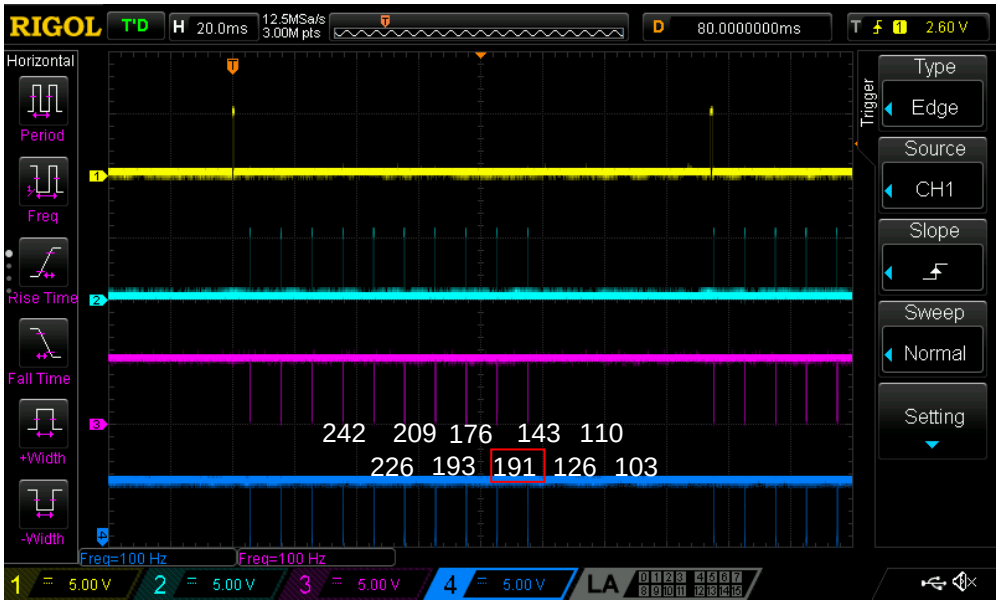

Figure 9

3.3.4 Workaround:

If the possibility of incorrect I2C data reads can't be accommodated for by external software checks, one can guarantee proper operation by stopping the CNT/DLY block's clock during I2C reads through one of the following methods: by disabling the oscillator block, by reconfiguring the CNT/DLY block's clock source, or by gating an external clock using a LUT (Look-up Table) in the signal matrix. After disabling the CNT/DLY block's clock, the count registers can be read without error. Please note that this workaround will add the I2C read and processing time to the counter's overall clock period.

The best workaround depends on the resource constraints of the application. If the oscillator block doesn't clock other logic elements within the design, a matrix output can be used to manually power down the oscillators for the I2C read. When the CNT/DLY block's clock source is routed internally from the oscillator block, I2C commands can temporarily reconfigure the CNT/DLY block's clock source registers to select "Ext. CLK. (From Matrix)." This action will disable the clock by connecting it to ground. If the CNT/DLY block is clocked from the signal matrix, a LUT can be used to gate the clock during an I2C read.

3.4 ACMP additional IN- leakage current

3.4.1 Effect

ACMP, PIN

3.4.2 Conditions

When all of the ACMPs are powered down.

3.4.3 Technical Description:

The SLG4653x have an additional leakage current through the PIN connected to the ACMP IN- input when all of the ACMPs are powered down. Typically, leakage through the PIN connected to IN- is much less than 1 µA. But when the ACMP is powered down and voltage is applied to the PIN, the leakage current may grow up to several µA (depending on the VDD and voltage applied).

3.4.4 Workaround:

Currently there is no workaround for this issue.

3.5 I2C Access to the 16-Bit Counter Data Registers

3.5.1 Effect

I2C, CNT0/DLY0/FSM0 & CNT1/DLY1/FSM1

3.5.2 Conditions

During I2C write/read while Counter is being clocked.

3.5.3 Technical Description

With regards to the GreenPAK counter registers, there are two specific I2C register sets to consider: the Counter Data registers and the Current Count Value registers. During underflow, overflow, and set events, the Counter Data register contents are loaded into a DFF chain that comprises the counter's internal structure. The Current Count Value registers are "Read Only" registers that output directly from this DFF chain.

Case 1: "Stop and Restart" bit disabled.

During normal operation, the Counter Data registers can be re-written using I2C. The I2C controller loads data into the 16-bit counter block using sequential 8-bit packets.

The timing diagram in Figure 10 sets up a case where an improper output will be seen on the counter. In this example, the initial Counter Data value (D) is set up as 0x00FF. As you can see, the Current Count Value (Q) decrements from 255 down to 0 with rising edges on CLK. At time “1” denoted by a red number 1, an I2C command (not shown) has just been sent to the GreenPAK device to reconfigure the Counter Data value to 0x0101.

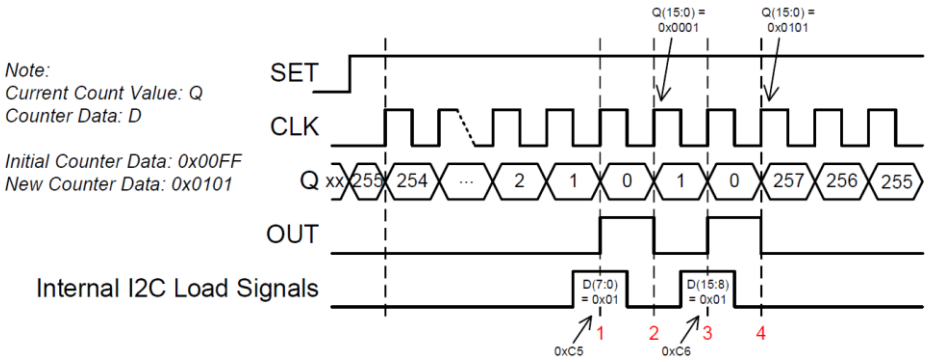

Figure 10: I2C Write of 16-Bit Counter Data Example

As previously mentioned, the 16-bit value is loaded from the I2C buffers to the Counter Data registers in 8-bit chunks. Right at time “1”, the LSB byte (0x01) is loaded into the Counter Data registers. This makes the Counter Data value temporarily equal to 0x0001. At time “2” between the internal I2C load signals, the counter receives a rising edge on its clock input. This causes the Current Count Value to underflow and to be loaded with the Counter Data value which is 0x0001. At time “3”, the MSB byte (0x01) is loaded into the Counter Data registers setting the Counter Data value to 0x0101. At time “4”, the Current Count Value underflows again and is set to the Counter Data value of 0x0101.

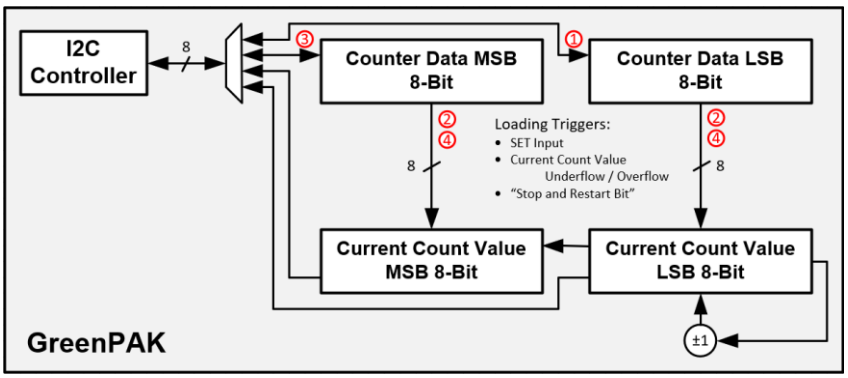

Figure 11: 16-Bit I2C Counter Block Diagram

This behavior results in two output pulses on OUT when only one pulse is expected. Observing this behavior depends upon the asynchronous timing between the counter’s clock and the I2C write command to the Counter Data registers. For some applications, the extra pulse on OUT can be ignored and the counter will self-correct after underflowing, overflowing, or being SET/RESET through the GreenPAK connection matrix.

Case 2: “Stop and Restart” bit enabled.

When the “Stop and Restart” bit is enabled, the counter’s clock is gated internally to stop the counting operation if the Control Data registers are being read from and written to via I2C. This bit was added to accommodate for the I2C write behavior described above. Although it solves the I2C

problem of writing to the Counter Data registers while clocking, it introduces some additional undesired behavior.

As the design has currently been implemented, the blocking feature for the counter's clock is triggered by monitoring for the Control Data register addresses in the "Word Address" portion of the I2C command structure. Since the design only looks at the address, both I2C reads and writes will gate the counter's clock for a period of about Tstop. Under most circumstances, temporarily gating the counter's clock won't impact the overall behavior of the counter excluding the loss of a few clock edges during the ~45µs window. (Assuming a 400kHz I2C Speed)

$$T_{stop} \approx 2 * 9 * \frac{1}{f_{CLK_{I2C}}}$$

For the following condition, the Counter Data register contents are unexpectedly loaded into the counter's DFF chain. This behavior occurs when there is a falling edge on the counter block's clock input during the data return portion of an I2C read of the Counter Data registers. Figure 10 shows this falling edge boxed in light blue followed by two I2C reads of the Current Count Value registers.

DIO0 – External Clock for Counter Block
DIO3 – MCU trigger for Scope Capture

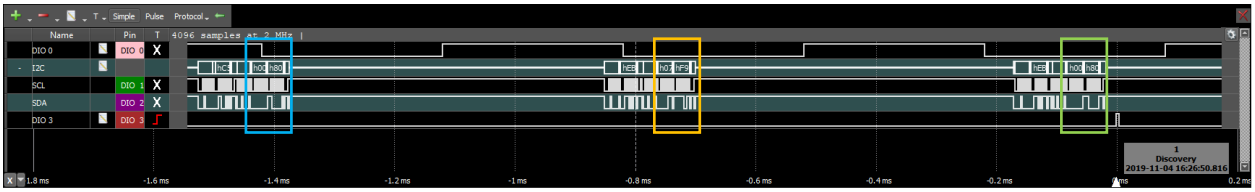

Figure 12: Counter Behavior

As you can see by the MCU data log below, the Current Count Value gradually increases from 0xF904 to 0xF907 but is unexpectedly reset to 0x8000. Please note that the colored HEX values in the MCU Log correspond to the boxed I2C read commands shown in the waveform above. As previously described, this reset occurs because there is a falling edge on the counter's clock input while the GreenPAK device is responding to an I2C read request to 0xC5 and 0xC6. (Note that 0xC5/0xC6 are the Counter Data registers and 0xEB/0xEC are the Current Count Value registers for CNT0.)

~~~~~~~~~~ MCU I2C Read Log ~~~~~~~~~~

Current Count Value (0xEB, 0xEC): 0xF904
Current Count Value (0xEB, 0xEC): 0xF905
Current Count Value (0xEB, 0xEC): 0xF906
Counter Data (0xC5, 0xC6): 0x8000
Current Count Value (0xEB, 0xEC): 0xF907
Current Count Value (0xEB, 0xEC): 0x8000
Current Count Value (0xEB, 0xEC): 0x8001
Current Count Value (0xEB, 0xEC): 0x8003

The reason that the counter isn't set to 0x8000 on the I2C read immediately following the Counter Data query is because the counter requires two rising edges on its clock input to load the data from the Counter Data registers into the DFF chain. Note that all rising edges during Tstop will be gated internally by the "Stop and Restart" feature.
~~~~~~~~~~

3.5.4 Workaround:

The workaround for this behavior depends upon the application and the types of I2C access required for the Counter Data and Current Count Value registers.

For applications where the application processor needs to read the Current Count Value while the counter is clocking, disable the “Stop and Restart Bit.” When this bit is disabled, the counter’s clock won’t be gated internally which avoids the loss of count values during Tstop. In order to guarantee the current count value after an I2C write to the Counter Data registers, the clock should be manually stopped for the I2C write or a SET/RESET command should be generated after the I2C write. (Case 1)

I2C Read of Counter Data: All I2C reads will return the correct value.

I2C Write of Counter Data: Accommodate for the double pulse described in Case 1.

Disable the counter’s clock during the I2C write.

Manually SET/RESET the counter after the I2C write is complete.

Disregard the first set of pulses on the counter output after an I2C write by waiting for one counter cycle to complete. This allows the counter’s DFF chain to be loaded with the new Counter Data value after the counter overflows/underflows.

I2C Read of Current Count Value: All I2C reads will return the correct value.

For systems that require I2C write access to the Counter Data registers while the counter is clocking, enable the “Stop and Restart Bit.” (Case 2)

I2C Read of Counter Data: Avoid reading Counter Data via I2C while the counter is being clocked.

Within the MCU, we recommend saving the Counter Data register configuration as a software variable so that the Counter Data register doesn’t need to be read during counter operation.

I2C Write of Counter Data: All I2C writes will execute correctly.

Note that the counter will lose track of all rising edges on its clock input during the clock gating period following an I2C write to the Counter Data registers.

I2C Read of Current Counted Value: All I2C reads will return the correct values.

3.6 GPO output latched from I2C interface activity without slave address or I2C write operation match when IO Latching is Enabled

3.6.1 Effect

I2C, PINs

3.6.2 Conditions

When IO Latching is Enabled

3.6.3 Technical Description

The GreenPAK device does not check for a matching I2C Slave address condition before triggering ‘IO Latching’. A Start bit on the I2C bus will trigger IO Latching. A Start bit is when I2C_SDA goes low first followed by I2C_SCL going low after.

Example applications where this is problematic:
- Other devices share the same I2C bus
- I2C bus is powered down while VDD is still high.

In the example below, we will be monitoring the OSC output, which is measured by a scope on Channel 3 via PIN17 (“OSC_OUT0”). The I2C bus will be forced low to create a START bit.

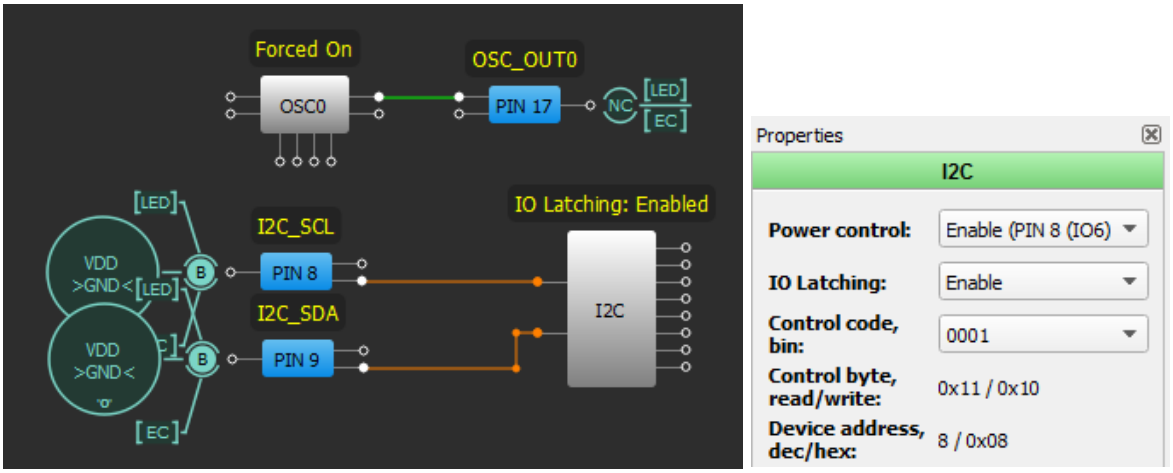

Figure 11

Channel 1 (yellow/top line) – PIN#9 (I2C_SDA)
Channel 2 (light green/2nd line) – PIN#8 (I2C_SCL)
Channel 3 (blue) – PIN#17 (OSC0_OUT0)

By toggling I2C_SDA low and then I2C_SCL low, the GreenPAK registers this as a start bit and latches all General Purpose Outputs such as PIN#17(“OSC_OUT0”). In the Figure 12, notice that PIN#17 (OSC_OUT0) stops toggling.

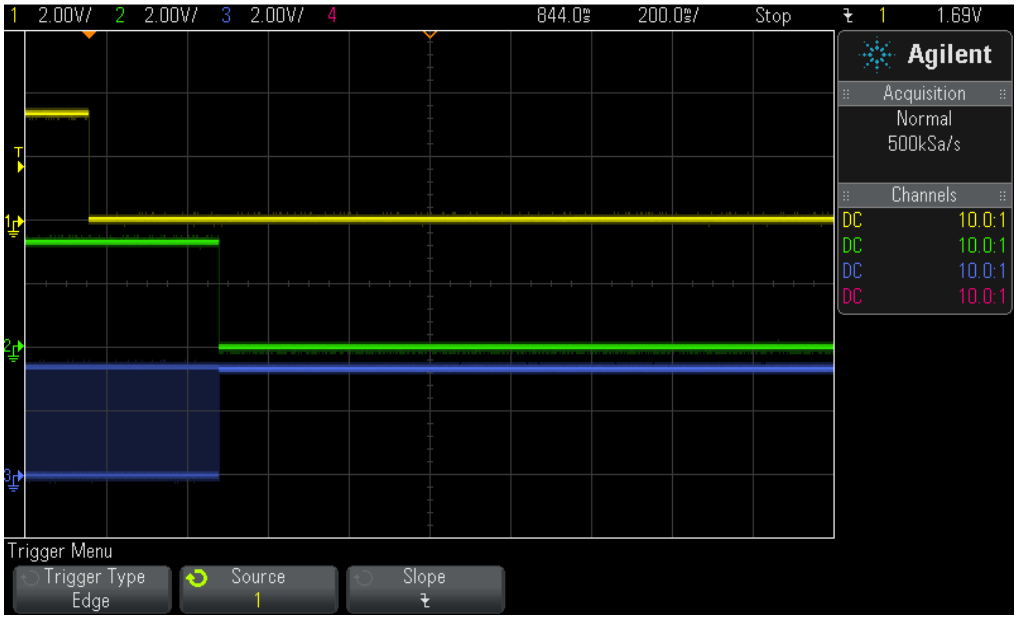

Figure 12

3.6.4 Workaround

Disable IO Latching in programmed OTP. This register is I2C-write locked during normal operation and therefore should be disabled in configuration file.
Do not share I2C bus with other devices and make sure the I2C bus is not powered down while the GreenPAK is still powered through VDD.

If the above cannot be satisfied, then consider using the SLG4658x devices, where this errata has been fixed.

Document Revision History

Revision	Date	Description
1.16	31-Mar-2020	Added issue: GPO output latched from I2C interface activity without slave address or I2C write operation match when IO Latching is Enabled
1.15	16-Dec-2019	Added issue I2C Access to the 16-Bit Counter Data Registers

SLG4653x Errata

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

Disclaimer

Contacting Dialog Semiconductor

United Kingdom (Headquarters)
Dialog Semiconductor (UK) LTD
Phone: +44 1793 757700

Germany
Dialog Semiconductor GmbH
Phone: +49 7021 805-0

The Netherlands
Dialog Semiconductor B.V.
Phone: +31 73 640 8822

Email:
enquiry@diasemi.com

North America
Dialog Semiconductor Inc.
Phone: +1 408 845 8500

Japan
Dialog Semiconductor K. K.
Phone: +81 3 5769 5100

Taiwan
Dialog Semiconductor Taiwan
Phone: +886 281 786 222

Web site:
www.dialog-semiconductor.com

Hong Kong
Dialog Semiconductor Hong Kong
Phone: +852 2607 4271

Korea
Dialog Semiconductor Korea
Phone: +82 2 3469 8200

China (Shenzhen)
Dialog Semiconductor China
Phone: +86 755 2981 3669

China (Shanghai)
Dialog Semiconductor China
Phone: +86 21 5424 9058