

SLG47525/28 Errata Note

## Abstract

This document contains the known errata for SLG47525/28 and the recommended workarounds.

## Contents

1. Information .....	1
2. Errata Summary .....	1
3. Errata Details.....	2
4. Revision History .....	14

## 1. Information

<b>Package(s)</b>	14-pin STQFN (SLG47525): 2.0 mm x 2.2 mm x 0.55 mm, 0.4 mm pitch
	20-pin STQFN (SLG47528): 2.0 mm x 3.0 mm x 0.55 mm, 0.4 mm pitch

## 2. Errata Summary

Issue #	Issue Title
1	<a href="#">Counter Incorrect Operation after the Reset</a>
2	<a href="#">FILTER Cell does not Filter Out Glitches</a>
3	<a href="#">Incorrect I2C Reads of the 8-bit Counter Registers</a>
4	<a href="#">ACMP Additional IN- Leakage Current</a>
5	<a href="#">I2C Access to the 16-Bit Counter Data Registers</a>
6	<a href="#">GPO Output Latched from I2C Interface Activity without Slave Address or I2C Write Operation Match when IO Latching is Enabled</a>
7	<a href="#">OSC Long Start-Up Time</a>

### 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.

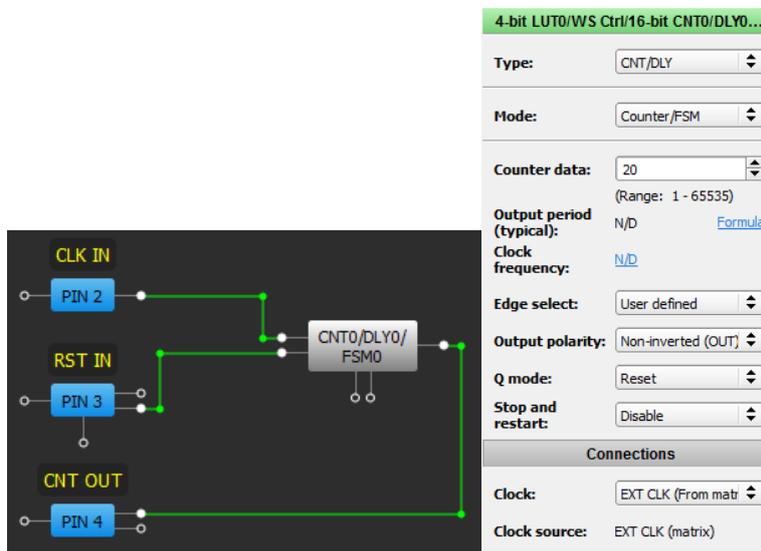


Figure 1



Figure 2

##### 3.1.4. Workaround

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

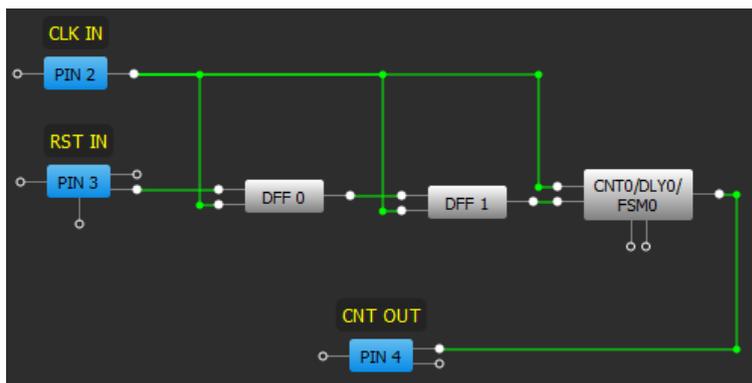


Figure 3

### 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 it out. There are several factors like input frequency, duty cycle, and LOW duration in such signal that may lead to its passing through FILTER block.

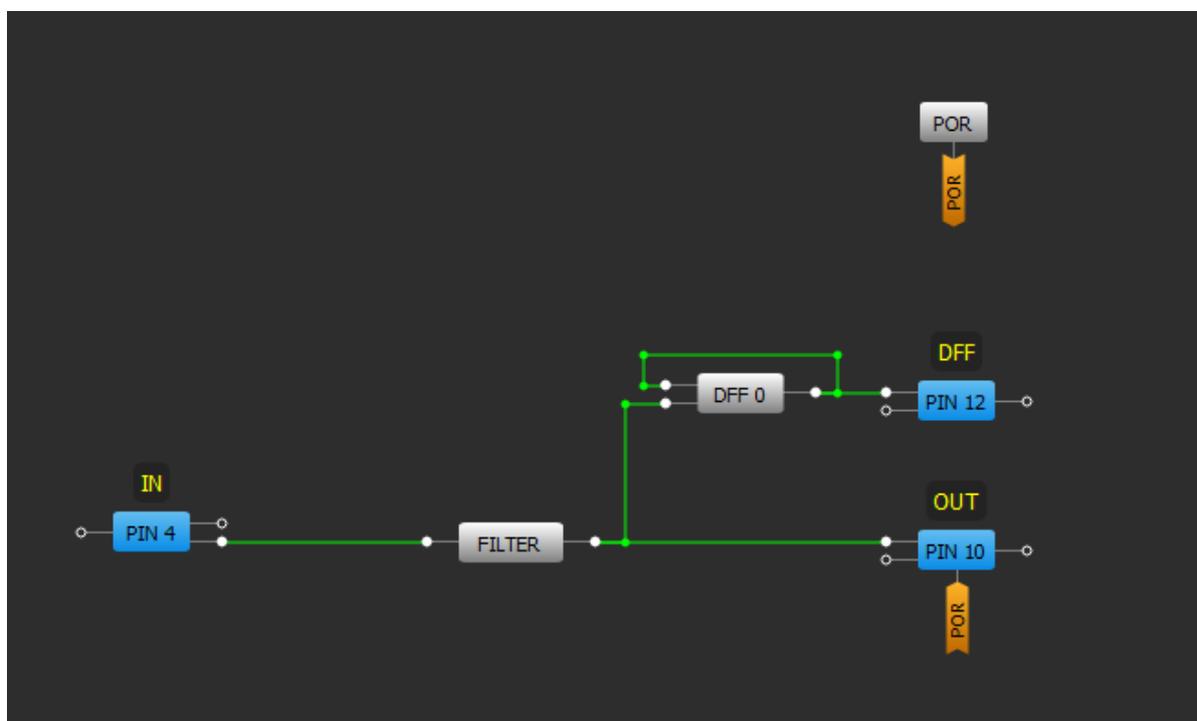


Figure 4

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

Channel 2 (light blue/2<sup>nd</sup> line) – PIN#10 (OUT).

Channel 3 (magenta/3<sup>rd</sup> line) – PIN#12 (DFF).

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



Figure 5

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



Figure 6

- Period is 60 ns. Pulse width is 30 ns, DC = 50 % (incorrect functionality).



Figure 7

- Period is 60 ns. Pulse width is 40 ns, 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 I<sup>2</sup>C 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 I<sup>2</sup>C LATCH signal, when the I<sup>2</sup>C LATCH signal and the clock input occur at about the same time.

#### 3.3.3. Technical Description

Asynchronous interaction between the CNT/DLY clock input and the I<sup>2</sup>C LATCH signal (generated by an I<sup>2</sup>C read command of the CNT/DLY block's count value) can result in an incorrect I<sup>2</sup>C data read. The CNT/DLY block will count accurately, but the count value transferred into the block's I<sup>2</sup>C read register might be loaded incompletely if the I<sup>2</sup>C LATCH signal and the clock input occur at about the same time.

The example data capture in Figure 9 shows ten periodic I<sup>2</sup>C 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 I<sup>2</sup>C 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/2<sup>nd</sup> line) – PIN#1 (I<sup>2</sup>C Read Triggers).

Channel 3 (magenta/3<sup>rd</sup> line) – PIN#8 (I<sup>2</sup>C SCL).

Channel 3 (dark blue/4<sup>th</sup> line) – PIN#9 (I<sup>2</sup>C SDA).

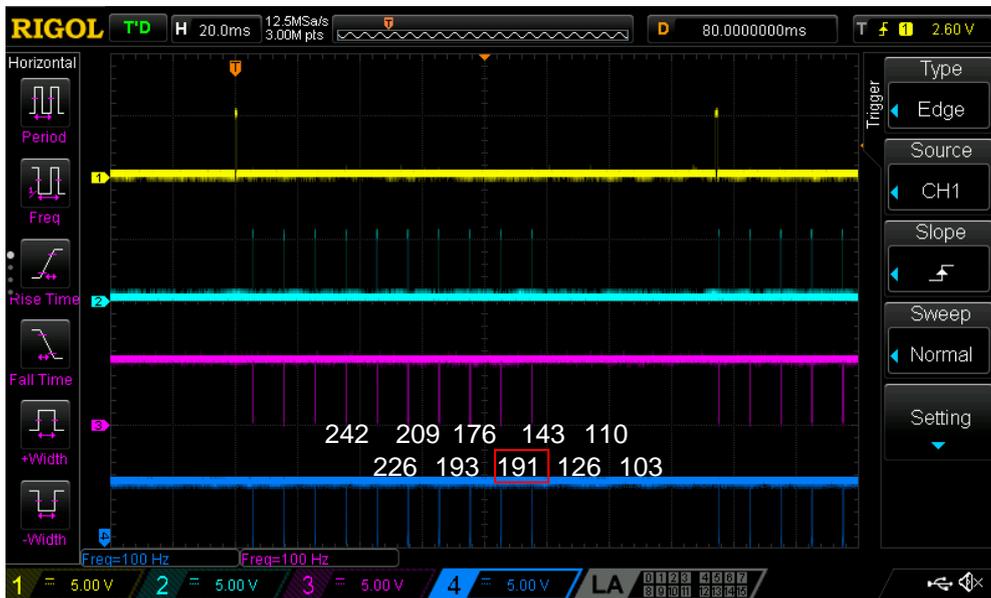


Figure 9

#### 3.3.4. Workaround

If the possibility of incorrect I<sup>2</sup>C data reads cannot be accommodated for by external software checks, one can guarantee proper operation by stopping the CNT/DLY block's clock during I<sup>2</sup>C 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 I<sup>2</sup>C 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 does not clock other logic elements within the design, a matrix output can be used to manually power down the oscillators for the I<sup>2</sup>C read. When the CNT/DLY block's clock source is routed internally from the oscillator block, I<sup>2</sup>C 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 the I<sup>2</sup>C 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 SLG47525/28 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  $\mu$ A. But when the ACMP is powered down and voltage is applied to the PIN, the leakage current may grow up to several  $\mu$ A (depending on the V<sub>DD</sub> and voltage applied).

### **3.4.4. Workaround**

Currently there is no workaround for this issue.

## **3.5 I<sup>2</sup>C Access to the 16-Bit Counter Data Registers**

### **3.5.1. Effect**

I<sup>2</sup>C, CNT0/DLY0/FSM0, and CNT1/DLY1/FSM1

### **3.5.2. Conditions**

During I<sup>2</sup>C write/read while Counter is being clocked.

### **3.5.3. Technical Description**

With regards to the GreenPAK counter registers, there are two specific I<sup>2</sup>C 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 I<sup>2</sup>C. The I<sup>2</sup>C 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 I<sup>2</sup>C command (not shown) has just been sent to the GreenPAK device to reconfigure the Counter Data value to 0x0101.

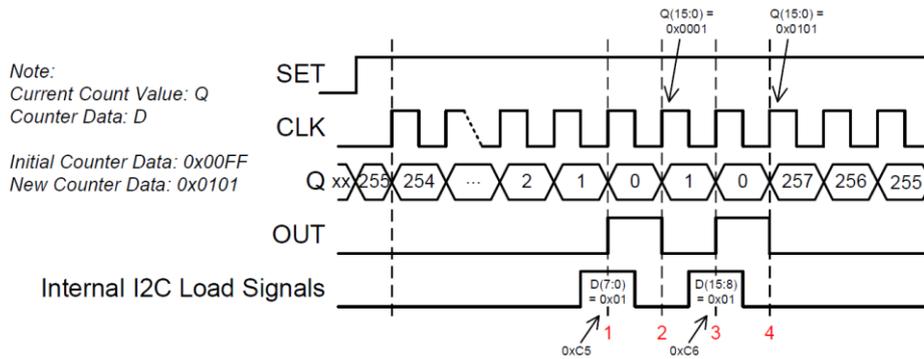


Figure 10. I<sup>2</sup>C Write of 16-Bit Counter Data Example

As previously mentioned, the 16-bit value is loaded from the I<sup>2</sup>C 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 I<sup>2</sup>C 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.

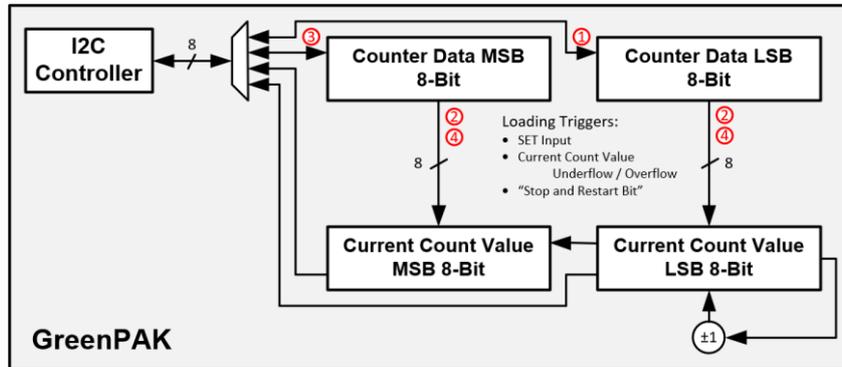


Figure 11. 16-Bit I<sup>2</sup>C Counter Block Diagram

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 an I<sup>2</sup>C 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 I<sup>2</sup>C. This bit was added to accommodate for the I<sup>2</sup>C write behavior described above. Although it solves the I<sup>2</sup>C 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 I<sup>2</sup>C command structure. Since the design only looks at the address, both I<sup>2</sup>C reads and writes will gate the counter’s clock for a period of about T<sub>stop</sub>. Under most circumstances, temporarily gating the counter’s clock will not impact the overall behavior of the counter excluding the loss of a few clock edges during the ~45 μs window (assuming a 400 kHz I<sup>2</sup>C 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 the I<sup>2</sup>C read of the Counter Data registers. Figure 12 shows this falling edge boxed in light blue followed by two I<sup>2</sup>C reads of the Current Count Value registers.

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

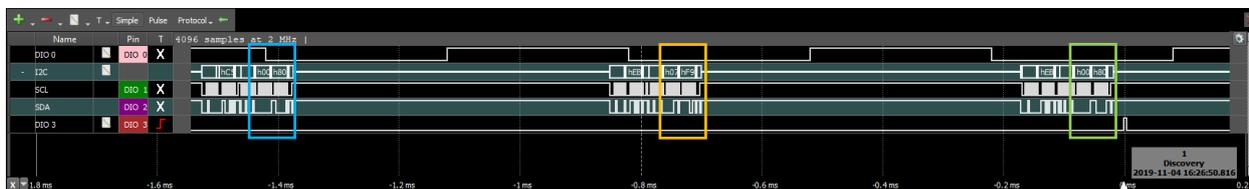


Figure 12. Counter Behavior

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 I<sup>2</sup>C 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 the I<sup>2</sup>C 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 is not set to 0x8000 on the I<sup>2</sup>C 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 T<sub>stop</sub> 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 I<sup>2</sup>C 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 will not be gated internally which avoids the loss of count values during T<sub>stop</sub>. In order to guarantee the current count value after the I<sup>2</sup>C write to the Counter Data registers, the clock should be manually stopped for the I<sup>2</sup>C write or a SET/RESET command should be generated after the I<sup>2</sup>C write (Case 1).

I<sup>2</sup>C Read of Counter Data: all I<sup>2</sup>C reads will return the correct value.

I<sup>2</sup>C Write of Counter Data: accommodate for the double pulse described in Case 1.

Disable the counter’s clock during the I<sup>2</sup>C write.

Manually SET/RESET the counter after the I<sup>2</sup>C write is complete.

Disregard the first set of pulses on the counter output after the I<sup>2</sup>C 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.

I<sup>2</sup>C Read of Current Count Value: all I<sup>2</sup>C reads will return the correct value.

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

I<sup>2</sup>C Read of Counter Data: avoid reading Counter Data via I<sup>2</sup>C 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 does not need to be read during counter operation.

I<sup>2</sup>C Write of Counter Data: all I<sup>2</sup>C 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 the I<sup>2</sup>C write to the Counter Data registers.

I<sup>2</sup>C Read of Current Counted Value: all I<sup>2</sup>C reads will return the correct values.

### 3.6 GPO Output Latched from I<sup>2</sup>C Interface Activity without Slave Address or I<sup>2</sup>C Write Operation Match when IO Latching is Enabled

#### 3.6.1. Effect

I<sup>2</sup>C, PINs

#### 3.6.2. Conditions

When IO Latching is enabled.

#### 3.6.3. Technical Description

The GreenPAK device does not check for a matching I<sup>2</sup>C Slave address condition before triggering ‘IO Latching’. A Start bit on the I<sup>2</sup>C 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 I<sup>2</sup>C bus
- I<sup>2</sup>C bus is powered down while V<sub>DD</sub> is still high.

In the example in [Figure 13](#), we will be monitoring the OSC output, which is measured by a scope on Channel 3 via PIN17 (“OSC\_OUT0”). The I<sup>2</sup>C bus will be forced low to create a START bit.

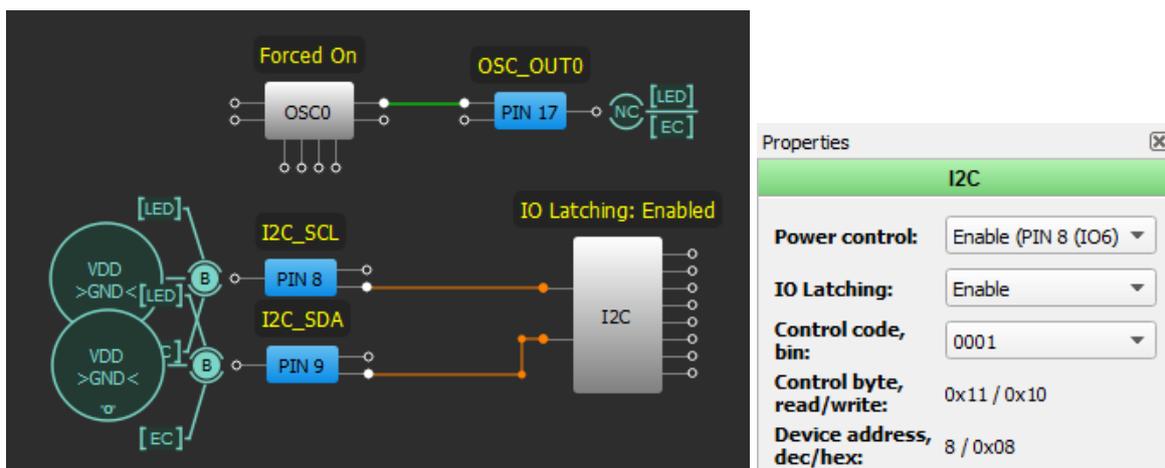


Figure 13

Channel 1 (yellow/top line) – PIN#9 (I2C\_SDA).

Channel 2 (light green/2<sup>nd</sup> 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 Figure 14, notice that PIN#17 (OSC\_OUT0) stops toggling.



Figure 14

### 3.6.4. Workaround

Disable IO Latching in programmed OTP. This register is I<sup>2</sup>C-write locked during normal operation and therefore should be disabled in configuration file.

Do not share I<sup>2</sup>C bus with other devices and make sure the I<sup>2</sup>C bus is not powered down while the GreenPAK is still powered through V<sub>DD</sub>.

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

## 3.7 OSC Long Start-Up Time

### 3.7.1. Effect

When a short pulse is applied to the PWR DOWN input, the oscillator exhibits a long start-up time.

### 3.7.2. Conditions

When a short pulse of less than ~40 ns is applied to the PWR DOWN input.

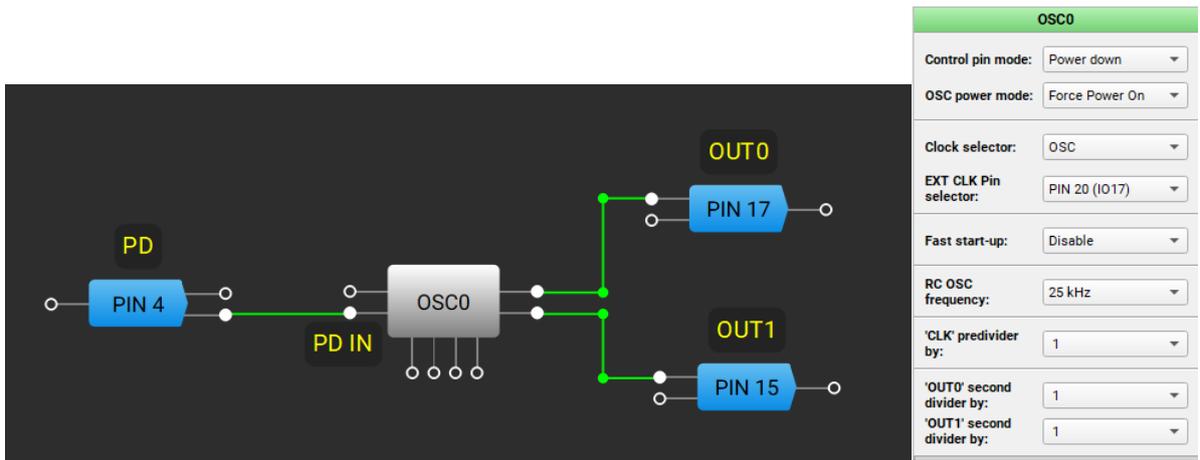


Figure 15. Test Design

### 3.7.3. Technical Description

If a pulse with a duration of less than ~40 ns is applied to the PWR DOWN input of the oscillator, the oscillator outputs remain stuck for milliseconds, as shown in Figure 16 and Figure 17. The issue is observed for all available oscillators.

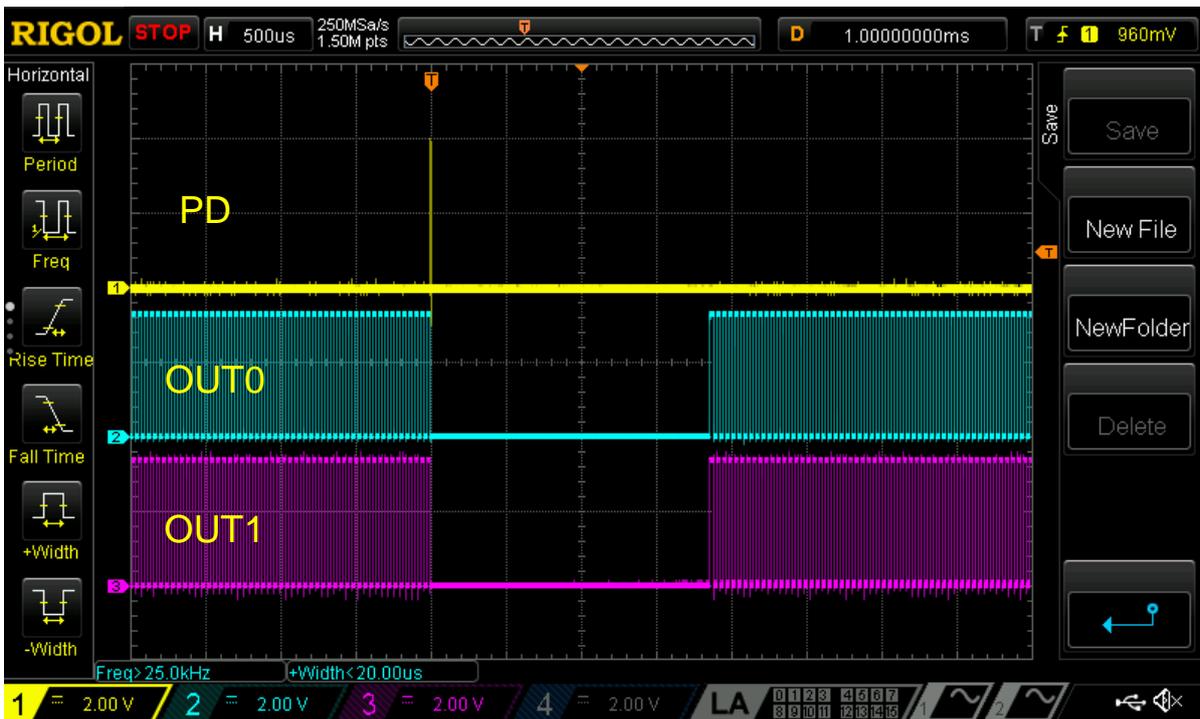


Figure 16. 25 kHz OSC Long Start-Up

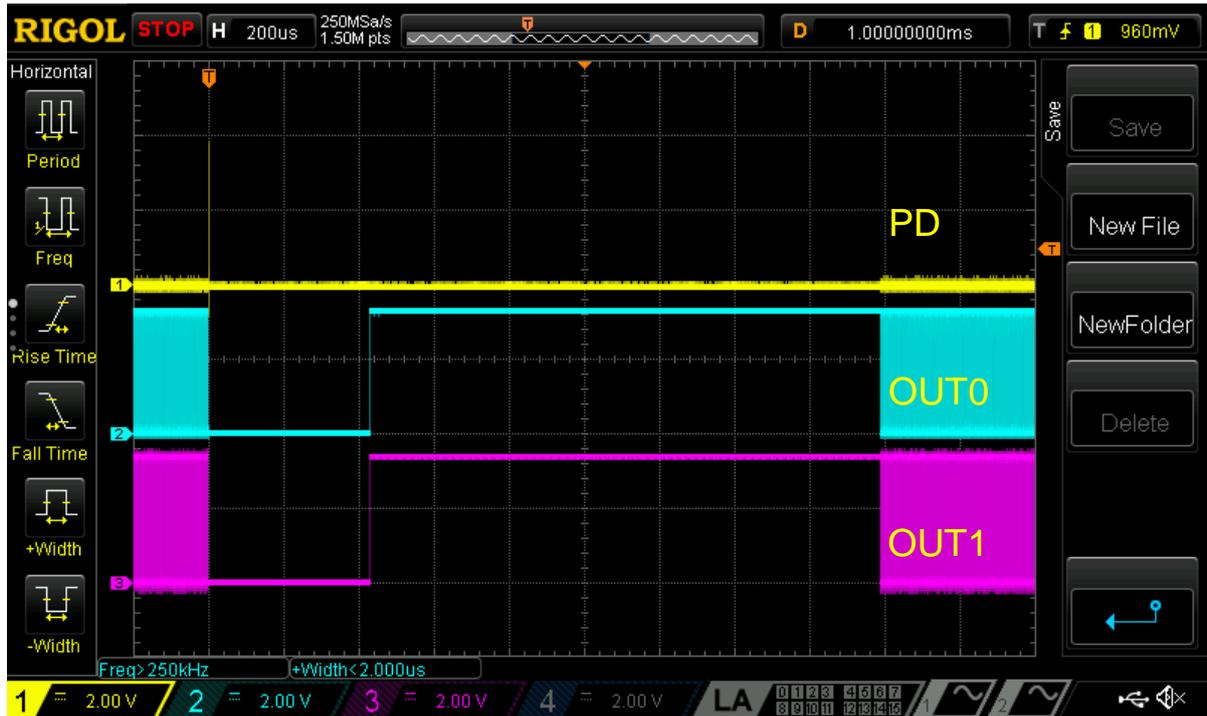


Figure 17. 2 MHz OSC Long Start-Up

### 3.7.4. Workaround

Change Fast start-up setting to Enable or omit applying glitches at PWR DOWN input in OSC.

OSC0	
Control pin mode:	Power down
OSC power mode:	Force Power On
Clock selector:	OSC
EXT CLK Pin selector:	PIN 20 (I017)
Fast start-up:	Enable
RC OSC frequency:	25 kHz
'CLK' predivider by:	1
'OUT0' second divider by:	1
'OUT1' second divider by:	1

Figure 18. Enabling Fast Start-Up to Eliminate OSC Start-Up Delay

## 4. Revision History

Revision	Date	Description
0.02	May 5, 2025	Added issue #7
0.01	Jun 3, 2024	Pre-production release

## IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

### Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan  
[www.renesas.com](http://www.renesas.com)

### Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

### Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit [www.renesas.com/contact-us/](http://www.renesas.com/contact-us/).