SLG4653x-E Errata Note

Abstract

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

Contents

1. Information

Package(s)	For SLG46533-E, SLG46537-E, SLG46538-E: 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-E, SLG46535-E, SLG46536-E: 14-pin STQFN: 2 x 2.2 x 0.55 mm, 0.4 mm pitch

2. 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 Target Address or I2C Write Operation Match when IO Latching is Enabled
7	Input Glitch Pattern Combination into DLYs Fails to Trigger Auto Power On of 25 kHz or 2 MHz OSC
8	OSC Long Start-Up Time
9	Long 2 MHz OSC Settling Time

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.

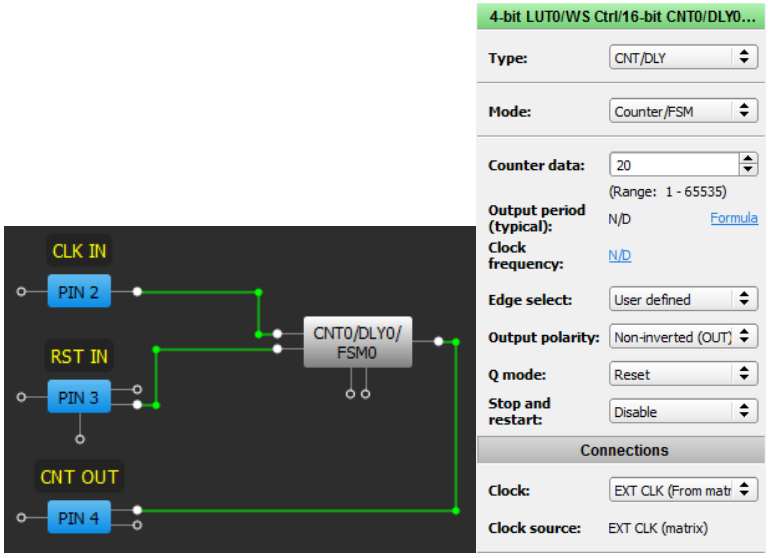

Figure 1

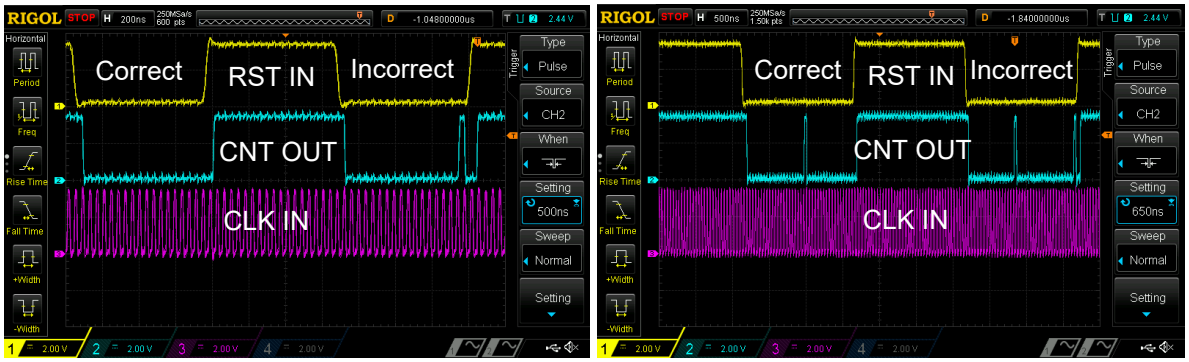

Figure 2

3.1.4. Workaround

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

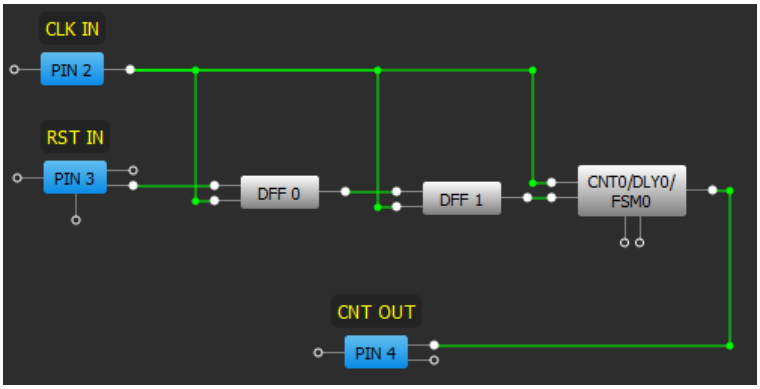

Figure 3

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.

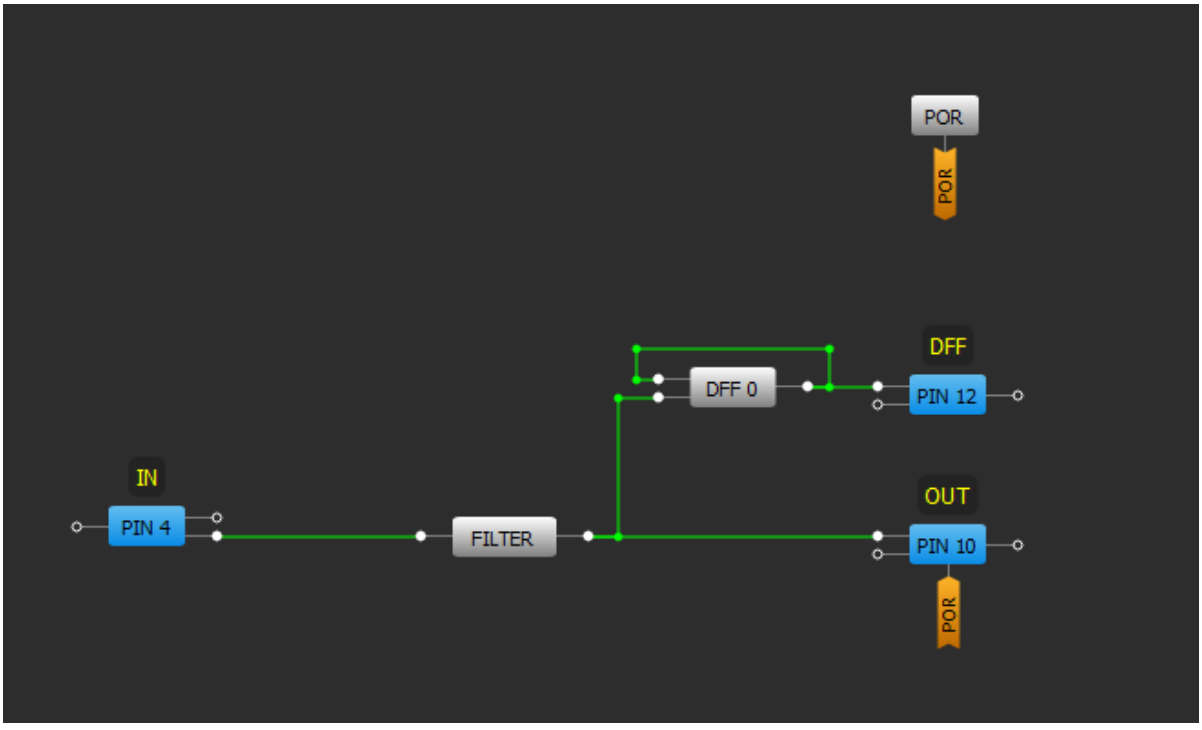

Figure 4

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

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

Channel 3 (magenta/3rd 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

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

Figure 7

4. 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 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. Technical 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 in Figure 9 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).

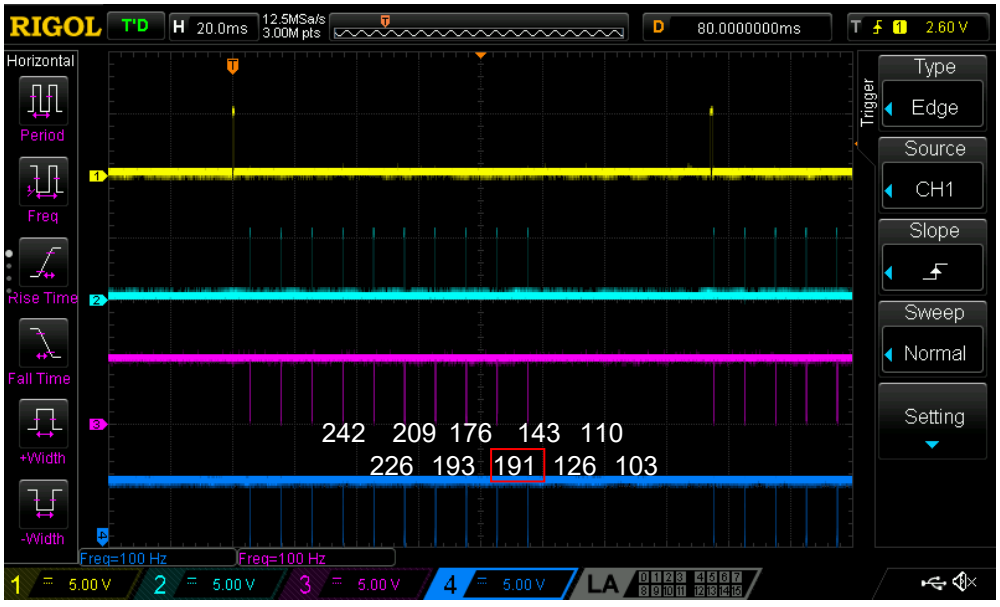

Figure 9

3.3.4. Workaround

If the possibility of incorrect I²C data reads cannot 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 does not 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 the 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 V_{DD} 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, and 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.

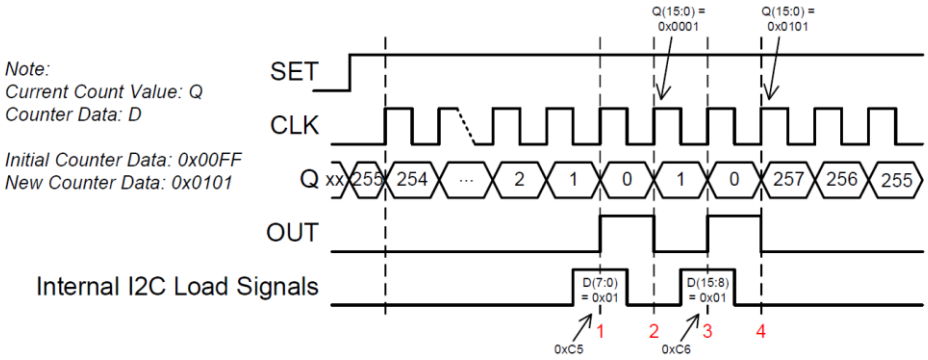

Figure 10. I²C Write of 16-Bit Counter Data Example

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.

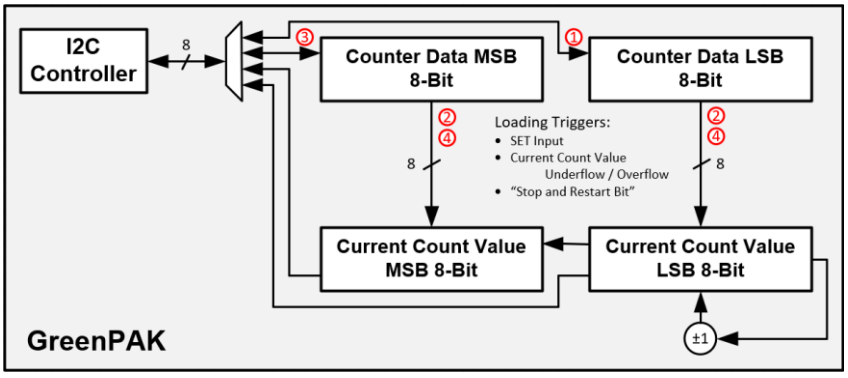

Figure 11. 16-Bit I²C Counter Block Diagram

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 an 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 T_{stop}. Under most circumstances, temporarily gating the counter’s clock will not impact the overall behavior of the counter exclu=ding the loss of a few clock edges during the ~45 µs window (assuming a 400 kHz 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 the I²C read of the Counter Data registers. Figure 12 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.

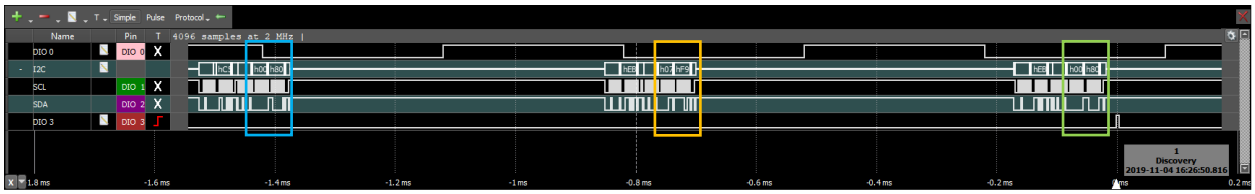

Figure 12. Counter Behavior

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 the 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 is not 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 T_{stop} 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 will not be gated internally which avoids the loss of count values during T_{stop}. In order to guarantee the current count value after the 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 the 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, it is recommended 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.

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 the 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 Target 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 Target 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 V_{DD} is still high.

In the example in Figure 13, it will being monitored 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.

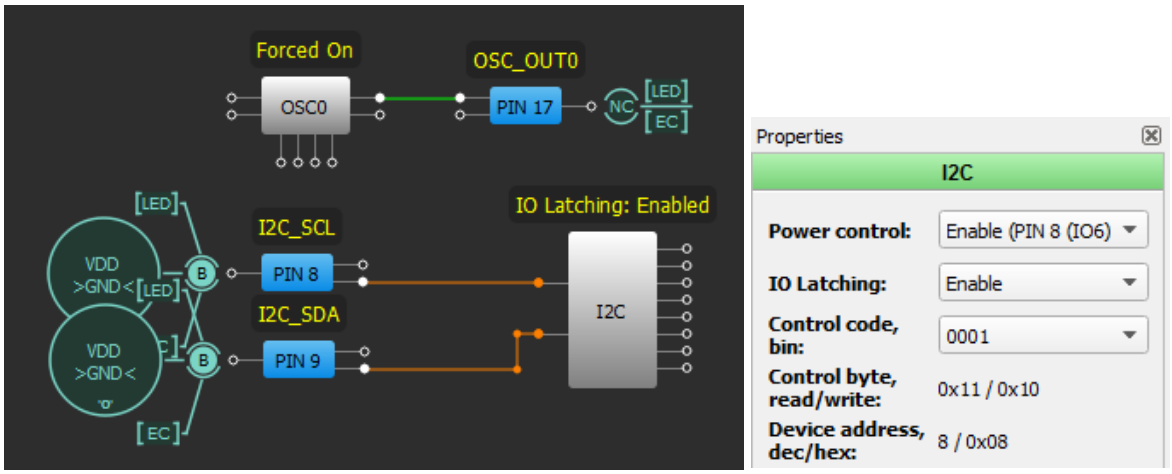

Figure 13.

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 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 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 V_{DD}.

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

3.7 Input Glitch Pattern Combination into DLYs Fails to Trigger Auto Power On of 25 kHz or 2 MHz OSC

3.7.1. Effect

Specific combinations of DLY inputs can fail to trigger Auto Power On to enable 25 kHz or 2 MHz OSC. Other OSC generators (25 MHz) do not have issue with Auto Power On setting.

3.7.2. Conditions

Auto Power On can potentially fail to enable OSC when all following conditions are present together:

1. 25 kHz or 2 MHz OSC is in Auto Power On mode.
2. DLYs are clocked by such OSC.
3. Input to one more such DLY have glitches < 200 ns.
4. OR of OSC trigger signals from all DLYs clocked by same OSC together form a long+short glitch pattern with precise (ns) timings as shown in Figure 15. The trigger signal generation per DLY is detailed in section 3.7.3.
5. During glitch period, no other DLY is active, meaning has already enabled OSC.

3.7.3. Technical Description

OSC generators have an Auto Power On mode which can be selected to automatically power on the OSC only when needed, such as when a DLY needs to count OSC cycles to time the delay output, thereby reducing quiescent power. Each individual DLY starts waiting in an inactive state, and when upon receiving an input edge (of polarity set by DLY configuration) then sets its individual trigger signal high. For example, in a rising-edge

DLY, a rising edge input sets this trigger high. For a falling-edge, the situation is inverted. In either case (or both edge DLY), the individual DLY trigger is reset if either an opposite edge is detected (therefore, canceling the DLY function) or the DLY finishes timing its output upon reaching the desired count per its setting. Therefore, an input pulse shorter than the DLY time is filtered out. The global Auto Power On circuitry then takes the OR of all the individual DLY triggers and subsequently sends the Controller enable signal to activate the OSC.

For this chip, the Auto Power On circuitry of the 25 kHz/2 MHz generators contains a circuit errata, which can potentially fail to power on the OSC for a specific pattern and timing of the OR of all related DLY trigger signals. The pattern is shown in Figure 15 (boxed in red) and consists of a relatively longer pulse (~145 ns) followed by a shorter (~5 ns) glitch of opposite polarity.

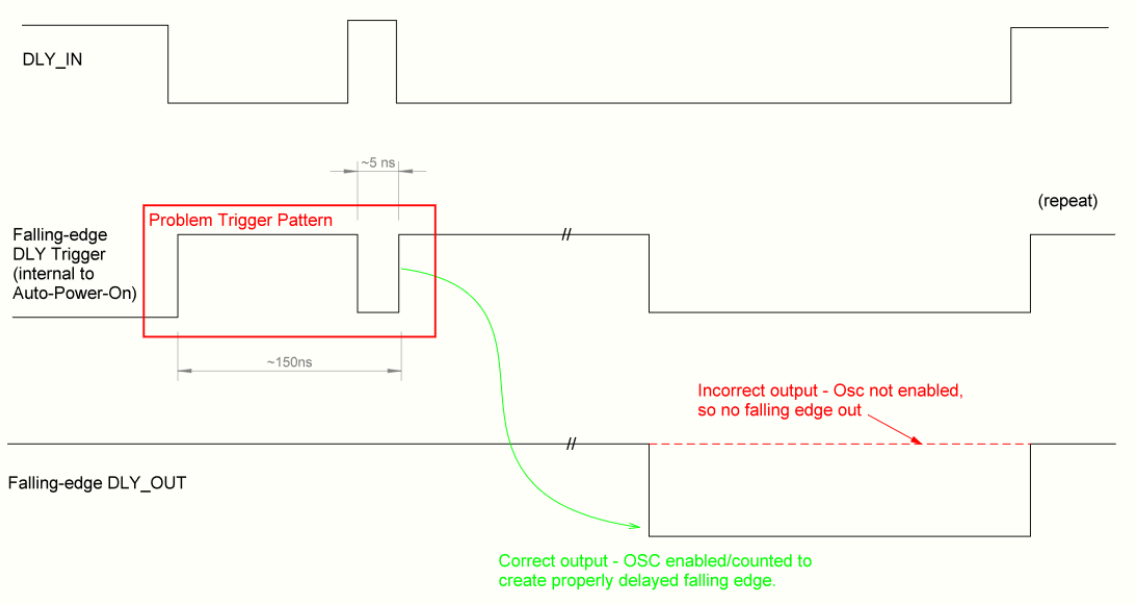

Figure 15. Problem Trigger Pattern

The error is difficult to capture as the timing must be exact within ns, and short (ns) pulses are difficult to generate glitch precise enough to induce the problem. GPIO naturally filter out ns pulses, so for purposes of errata capture, two simultaneous falling-edge DLY circuits (as shown in Figure 16) were used. By lining up two delays precisely at a particular timing relationship, it can be used the internal Auto Power On (OR logic) to generate the glitch pattern necessary to cause the error.

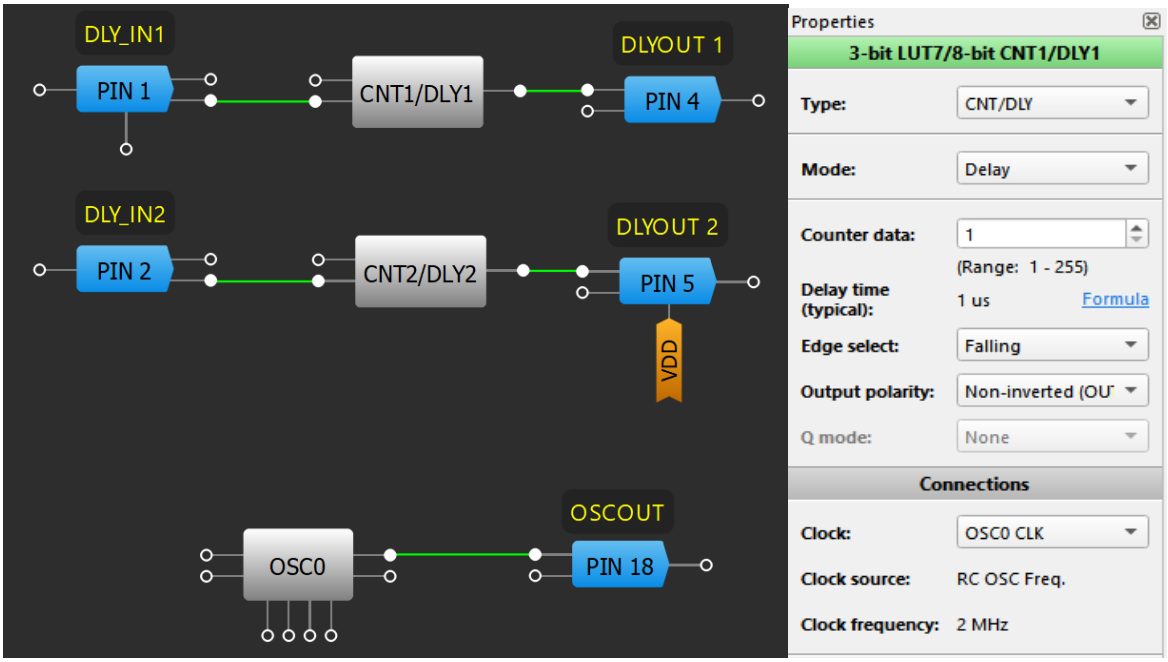

Figure 16. Test Circuit

Figure 17 shows a series of DLY output events are missing, where the OSC is correspondingly not triggering when it was supposed to. Figure 18 is a zoom in of the boxed region from Figure 17. The total glitch/chatter time was measured in this case at 152 ns. By using the composite OR of the two delay channels, it can be strobed asynchronously with tiny frequency variations, and so tune the timing to induce this errata. Note that the error is not persistent – the system is recovered when all DLYs are returned to inactive state, such as when input of falling edge DLY goes high, thus canceling delay, or after DLY out finishes.

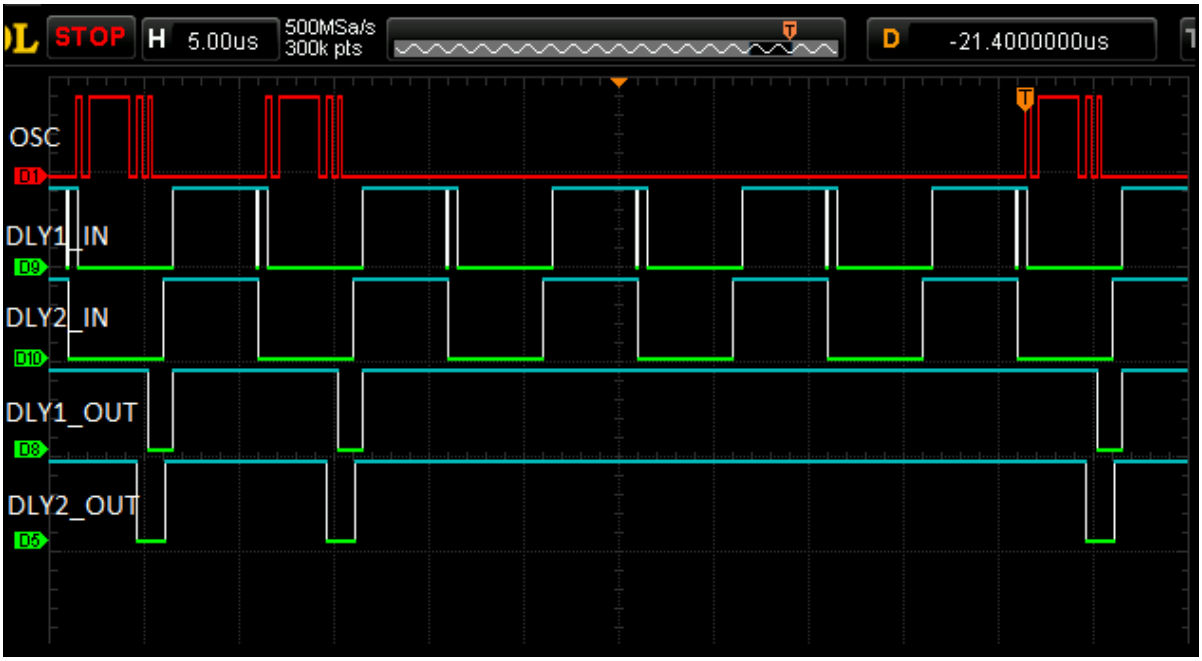

Figure 17. Errata Capture (Zoom Out)

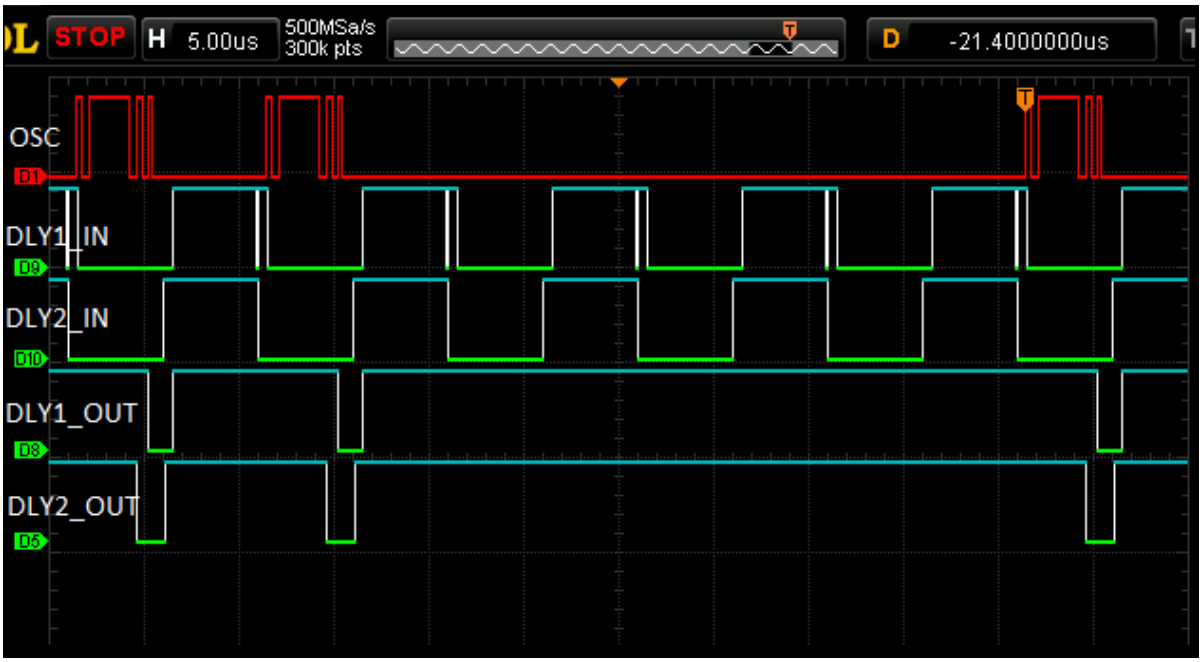

Figure 18. Errata Capture (Zoom In)

3.7.4. Workaround

Any one of the following prevents the issue:

1. Use signal conditioning circuits to prevent glitch on DLY inputs < 200 ns. Examples:
    a. ACMP with Hysteresis

b. External RC in front of Digital Input with Schmitt Trigger
c. Filter cell.

2. Set OSC power mode to Force Power On mode instead of Auto Power On.
3. Use different oscillator, such as 25 MHz Ring OSC, which does not have Auto Power On issue.

3.8 OSC Long Start-Up Time

3.8.1. Effect

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

3.8.2. Conditions

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

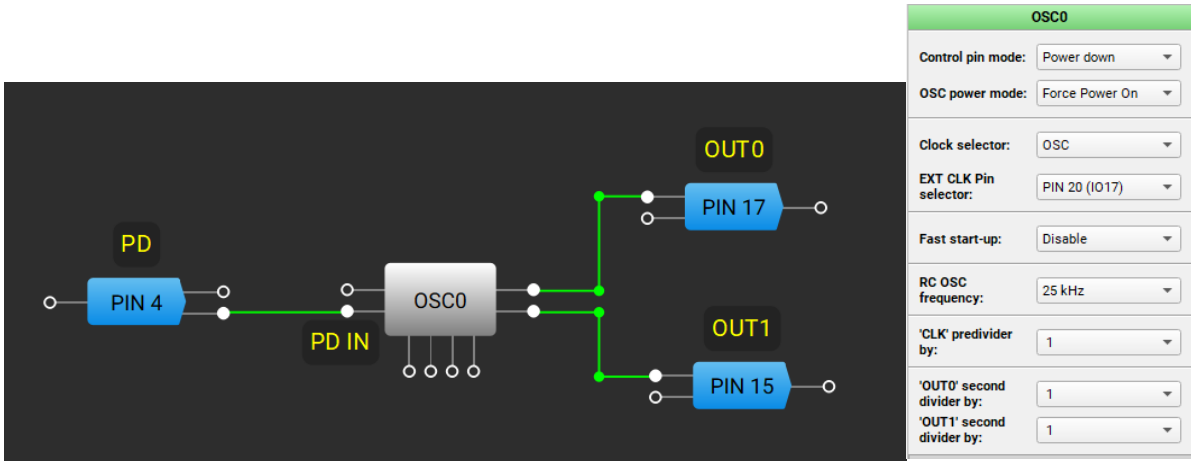

Figure 19. Test Design

3.8.3. 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 20 and Figure 21. The issue is observed for all available oscillators.

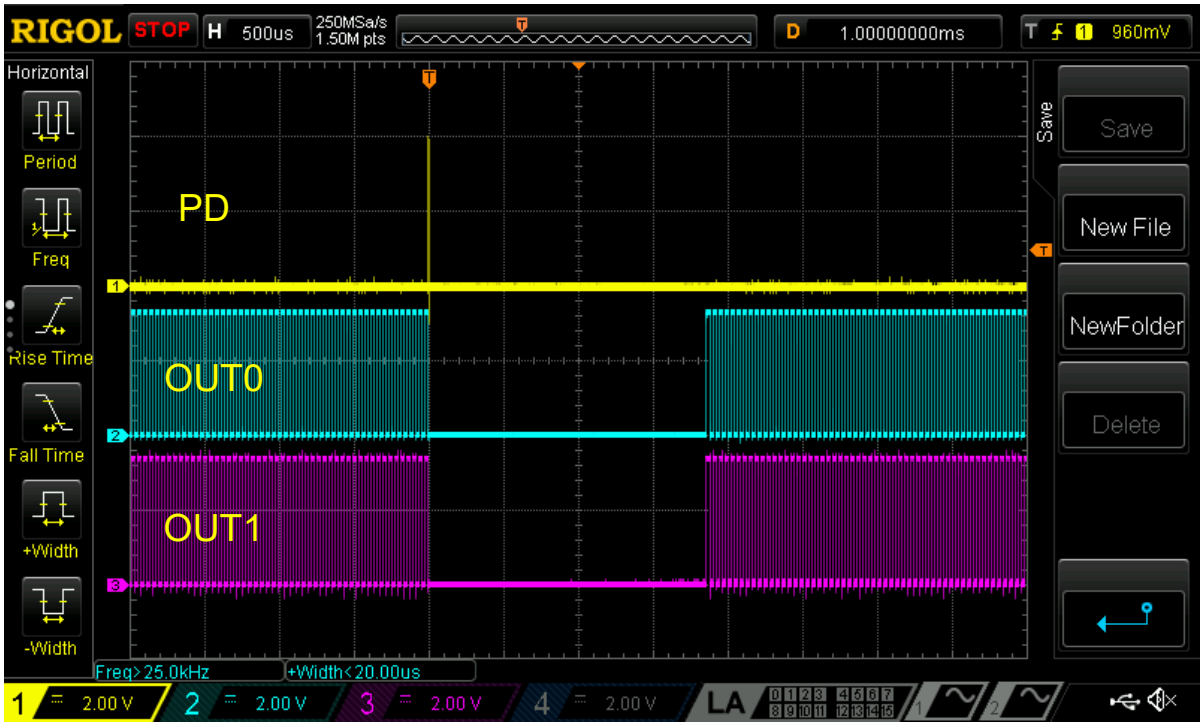

Figure 20. 25 kHz OSC Long Start-Up

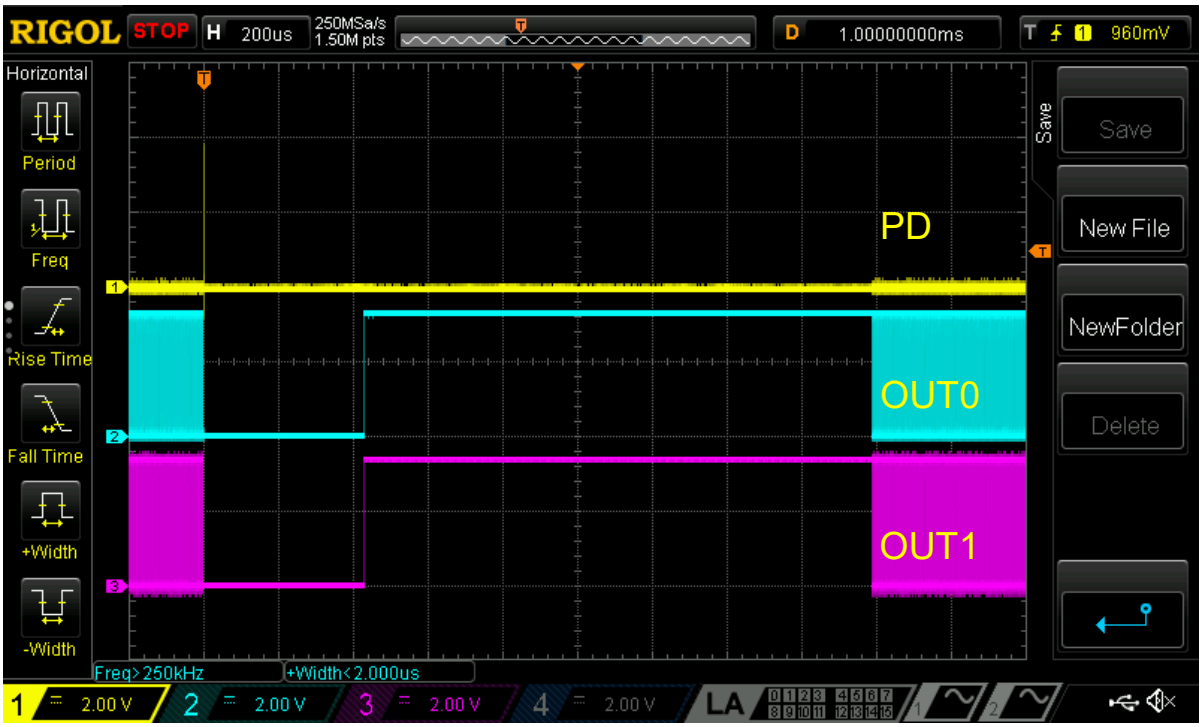

Figure 21. 2 MHz OSC Long Start-Up

3.8.4. Workaround

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

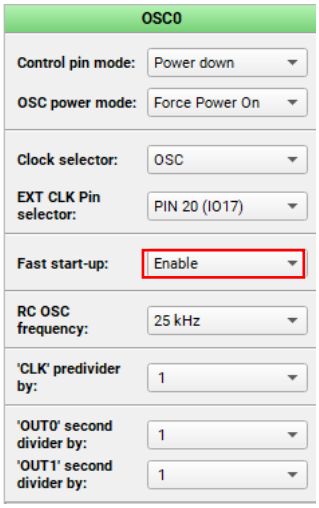

Figure 22. Enabling Fast Start-Up to Eliminate OSC Start-Up Delay

3.9 Long 2 MHz OSC Settling Time

3.9.1. Effect

OSC, Counter, Delay

3.9.2. Conditions

Fast Start-Up – Disable, Auto Power On

3.9.3. Technical Description

2 MHz OSC has an additional ~ 9 cycles settling period. Higher V_{DD} shows longer settling time.

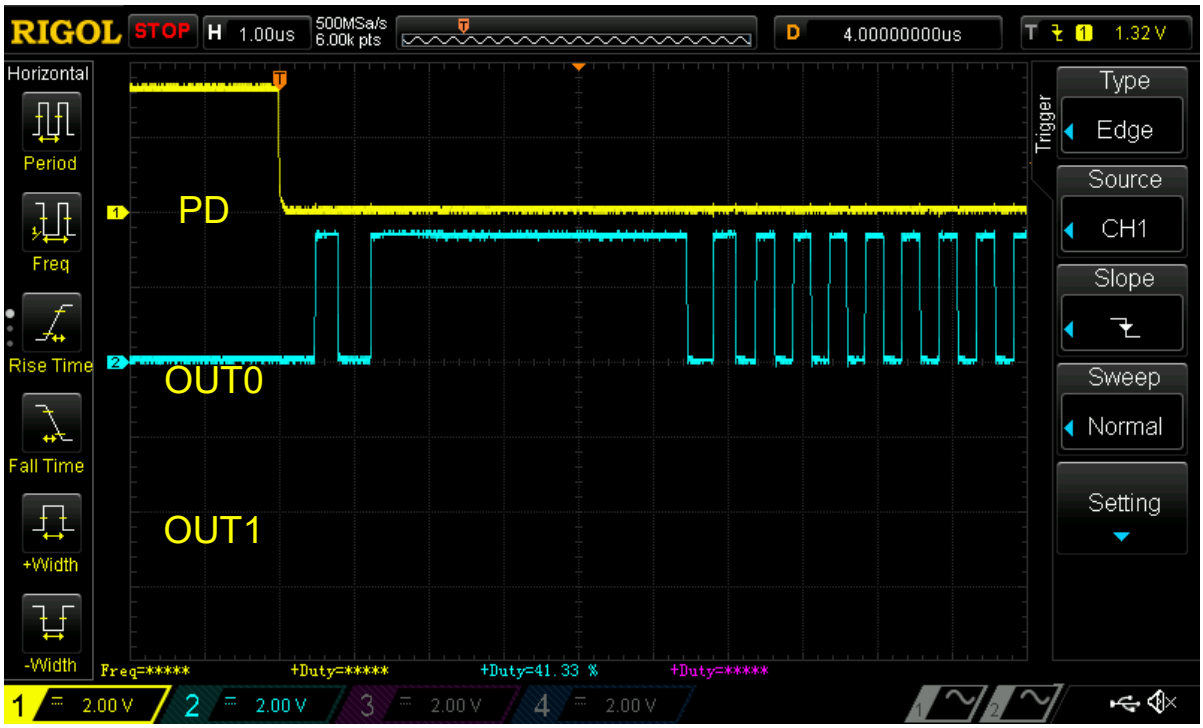

Figure 23. Channel 1 – OSC Power Down; Channel 2 –OSC Output

Such behavior will lead to substantial error in period calculations if the delay time is relatively small.

3.9.4. Workaround

Enable Fast Start-up option. Fast Start-up means forcing bias ready at the power-up instead of automatic enabling at OSC event. The standby current consumption difference between Fast start-up disabled and enabled is only an additional 300 nA.

Use the “Force power on” OSC power control option to make the OSC operate at all times. However, this will cause increased constant current consumption.

4. Revision History

Revision	Date	Description
1.00	Jul 24, 2025	Initial 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

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/.

Trademarks

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