Application Note

Bidirectional Counter (FSM)

AN-CM-286

Abstract

This application note shows how to build bidirectional counters having different interface methods designed in GreenPAK.

This application note comes complete with design files which can be found in the References section.

Contents

Figures

1 Terms and Definitions

FSM - Finite State Machine
SPI - Serial Peripheral Interface
I2C - Inter Integrated Circuit
DFF - D Flip-flop
LUT - Lookup Table
OVF - Overflow
OSC - Oscillator
IC - Integrated Circuit
DCMP - Digital Comparator
ZD - Zero Detector
CNT - Counter

2 References

For related documents and software, please visit:

GreenPAK™ Programmable Mixed-Signal Products | Renesas.

Download our free GreenPAK™ Designer software [1] to open the .gp files [2] and view the proposed circuit design. Use the GreenPAK development tools [3] to freeze the design into your own customized IC in a matter of minutes. Find out more in complete library of application notes [4] featuring design examples as well as explanations of features and blocks within the GreenPAK IC.

[1] Go Configure™ Software Hub | Renesas, Software Download and User Guide
[2] AN-CM-286 Bidirectional Counter.gp, GreenPAK Design File
[3] GreenPAK Development Tools, GreenPAK Development Tools Webpage
[4] GreenPAK Application Notes, GreenPAK Application Notes Webpage

Author: Ivan Vaskiv

3 Introduction

A counter is a digital circuit which is used for counting input events (pulses, edges). Digital electronic counters are usually built from a series of flip-flops connected in cascade.

This application note is intended to show how to build bidirectional counters (FSM) having different interface methods designed in GreenPAK. The methods to be implemented are:

16-bit FSM with SPI output

16-bit FSM with I2C output

FSM with parallel outputs (based on glue logic)

4 "16-bit FSM with SPI Output"

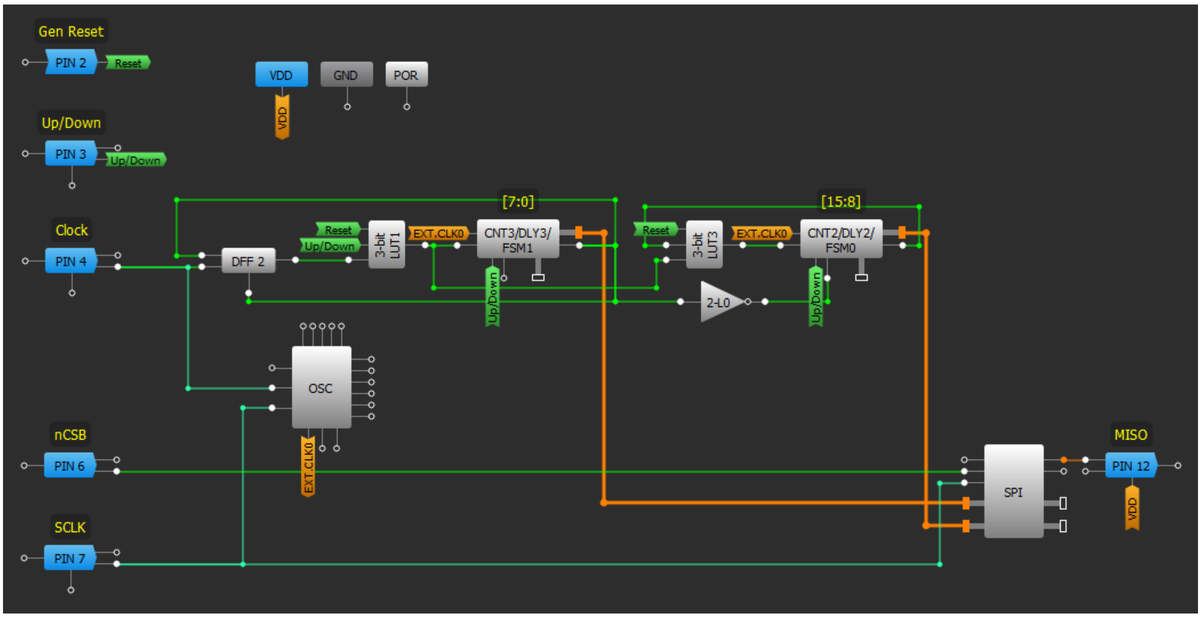

Figure 1: "16-bit FSM with SPI Output"

The "16-bit FSM with SPI output" counts input clocks in an internal 16-bit register (FSM0, FSM1), at any time a user can read the value via SPI, reset the 16-bit register, or change direction of counting. The design is implemented in SLG46140, or SLG46620 as well.

Let's see how it works. The 16-bit counter is implemented using two counters (FSM0 and FSM1 blocks) with additional logic, since SLG46140 doesn't have 16-bit FSM. Bits 15 – 8 are stored in FSM0, Bits 7 – 0 in FSM1. Both FSMs are connected to the SPI block, which can output serial data via SPI. The direction of counting is controlled by Up/Down pin. If this pin is HIGH, the system counts UP, if this pin is LOW, the system counts DOWN. Gen Reset pin is used to reset counter value (active HIGH).

Input clocks come from Clock input. These clocks are applied simultaneously at CLK inputs of FSM1 and FSM0. FSM1 counts each clock, whereas FSM0 counts only when FSM1 counter value is 255 and Up/Down signal is HIGH or when FSM1 counter value is 0 and Up/Down signal is LOW. This functionality is achieved using KEEP input of the FSM0. When this signal is HIGH the counter value of the FSM0 is not changing despite the clock signal. The KEEP input is connected to FSM1 output through an inverter. FSM1 output in turn is HIGH only when counter value is 0 and Up/Down signal is LOW, or when counter value is 255 and Up/Down signal is HIGH.

Since FSM counter value is reset to counter data when the FSM counter value reaches maximum (255) and FSM1 counter data is 255, FSM1 stops at a max value (255) when counting up and FSM1 UP input signal is HIGH. In order to avoid this situation, DFF2 and 3-bit LUT1 reset FSM1 by the next clock when FSM1 reaches max value and Up/Down is HIGH. If Up/Down is LOW, FSM1 is not reset. The situation is similar with FSM0. This one is also reset to counter data, and when FSM0 reaches max value (16383) it will then start to cause incorrect data. In order to avoid that situation, 3-bit LUT3 resets FSM0 by the next clock when both FSMs reach max value (SPI output code in this situation is 0xFFFF) and Up/Down is HIGH. Both FSMs are not reset if Up/Down is LOW.

D0 – PIN#6 (nCSB)

D1 – PIN#7 (SCLK)

D2 – PIN#12 (MISO)

D3 – PIN#2 (Gen Reset)

D4 – PIN#3 (Up/Down)

D5 – PIN#4 (Clock)

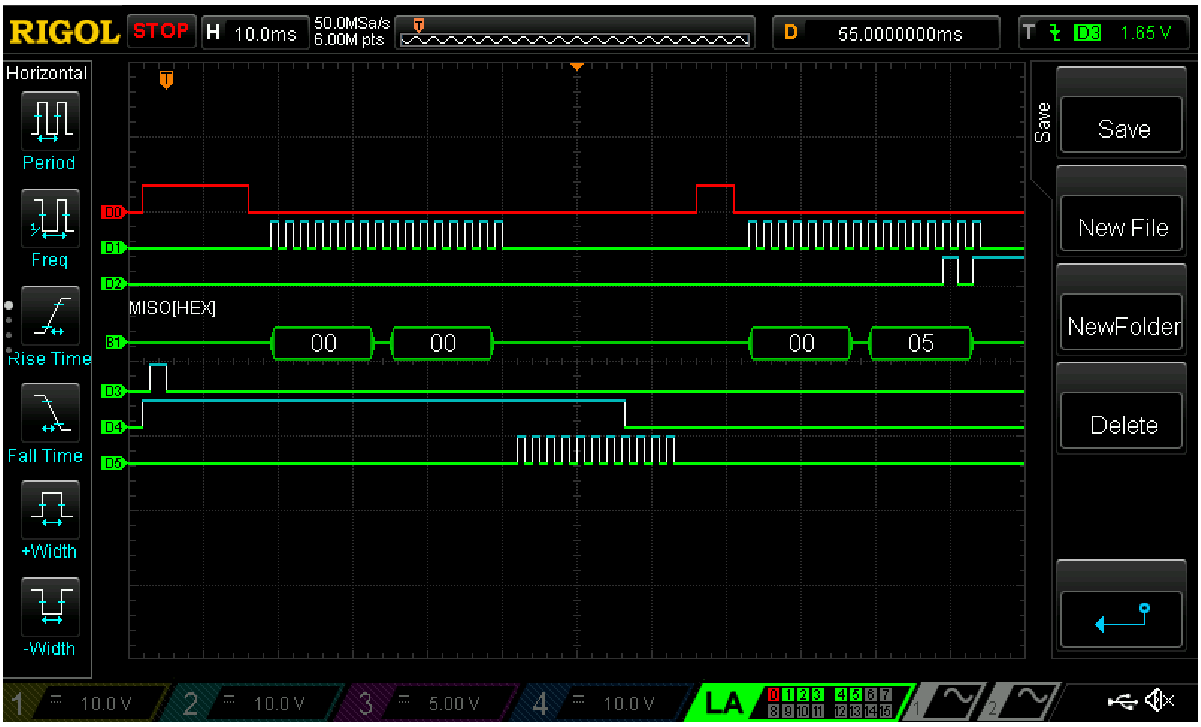

Figure 2: Functionality of “16-bit FSM with SPI Output”

The following design, which is shown on Figure 3, is just a modification of the 16-bit FSM with SPI output. This design can be used if we need to detect an overflow (OVF) event and set the initial value of the 16-bit counter (FSM0 and FSM1).

The initial value of the 16-bit counter is set to ½ maximum value (FSM0 data = 0x7F, FSM1 data = 0xFF) and can be adjusted easily. Initial value setting process takes ~9ms and starts just after the power up of GreenPAK IC, or when High pulse is applied to “Gen Reset” input. During those ~9ms FSM0 and FSM1 receive pulses from the internal OSC and will ignore pulses from Clock input. The initial value depends on CNT0 counter data and Pipe delay “OUT0 PD num” value. CNT0 counts the clock from internal oscillator simultaneously with 16-bit counter. Pipe Delay in turn counts output pulses from CNT0. When Pipe Delay OUT0 goes HIGH, the OSC is stopped by this signal (2-L5) and clock signal for 16-bit counter is taken from Clock input. To change the source of clocks, a 3-bit LUT0 is used. Also, when Pipe delay OUT0 is LOW, it forces the 16-bit counter to count DOWN, in order to guarantee the correct direction of counting during initial value's set process. The initial value is determined by the following formula:

$$Intial\ value = 65535 - ((CNT0\ Counter\ data + 1) * Pipe\ Delay\ OUT0\ value - 1)$$

In case of 16-bit counter overflow, an according OVF output will be set and latched. It will be reset after IC reset or when "Gen Reset" or "OVF Reset" signal comes. This function is implemented using the following components: DCMP0, DCMP1, 4-bit LUT0, 2-L4 and DFF2.

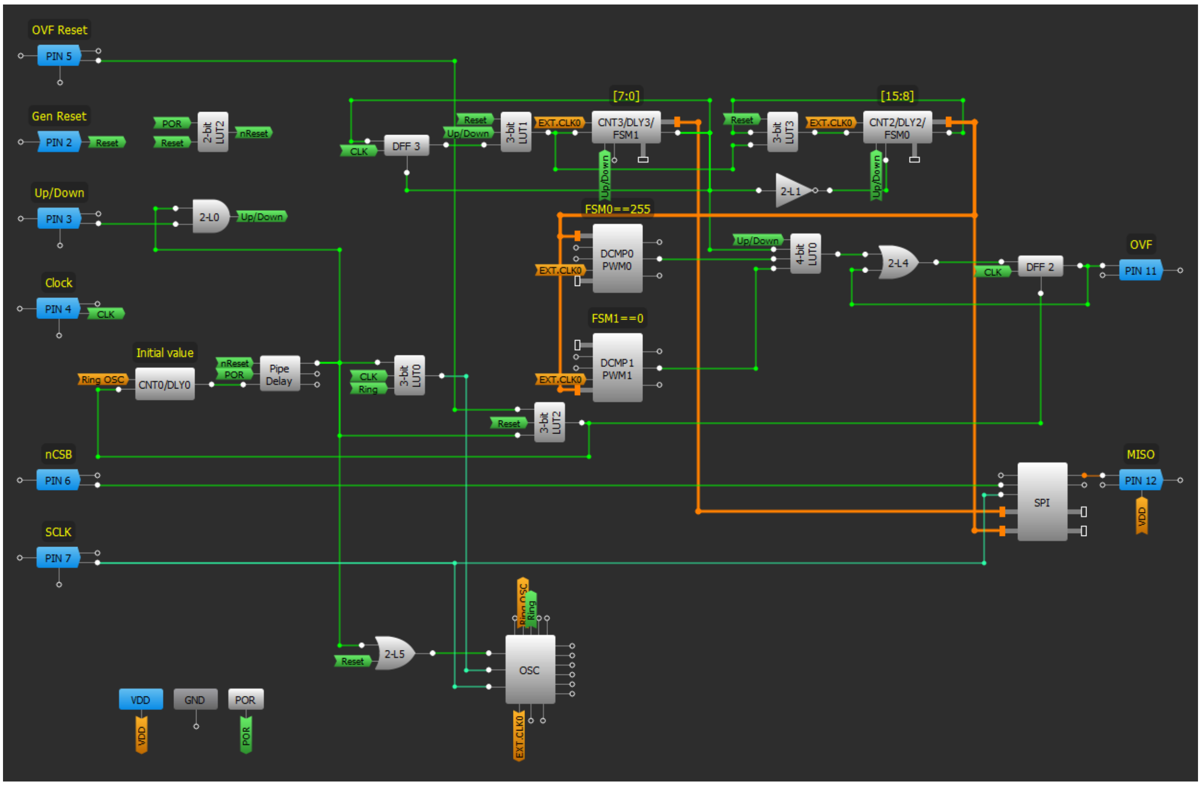

Figure 3: "16-bit FSM with SPI Output, OVF and Initial Value's Setting Functions"

D0 – PIN#6 (nCSB)
D1 – PIN#7 (SCLK)
D2 – PIN#12 (MISO)
D3 – PIN#5 (OVF Reset)
D4 – PIN#2 (Gen Reset)
D5 – PIN#3 (Up/Down)
D6 – PIN#4 (Clock)
D7 – PIN#11 (OVF)

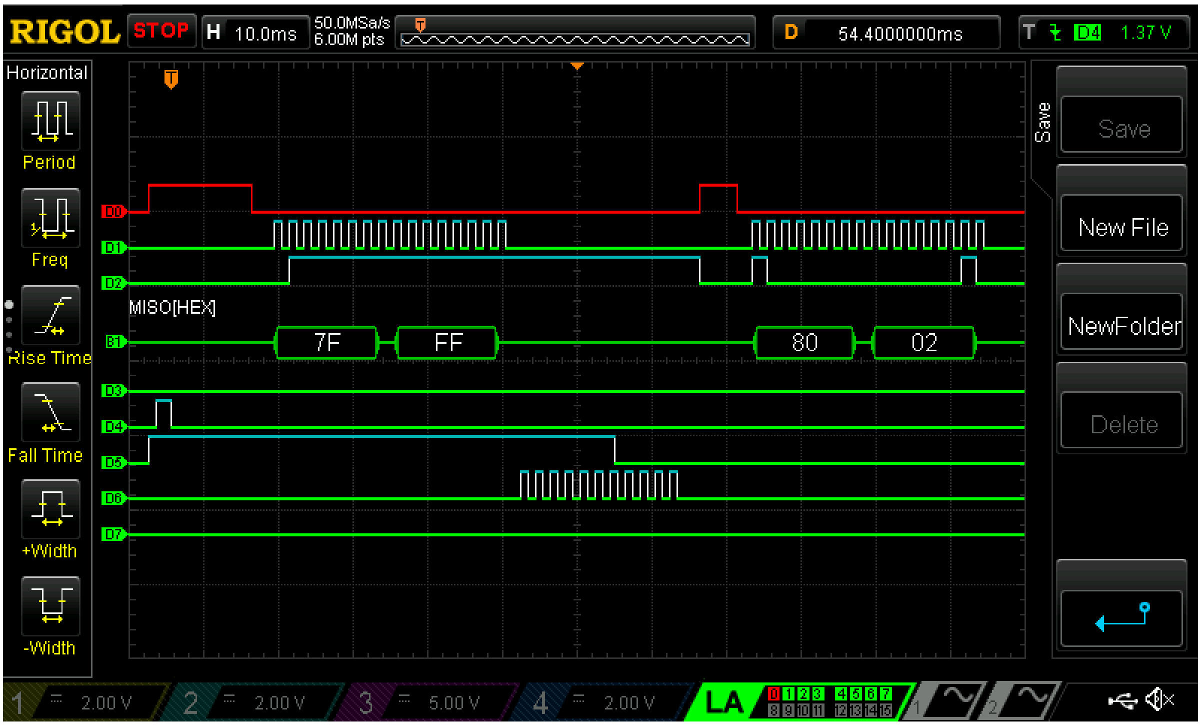

Figure 4: Functionality of "16-bit FSM with SPI Output, OVF and Initial Value's Setting Functions"

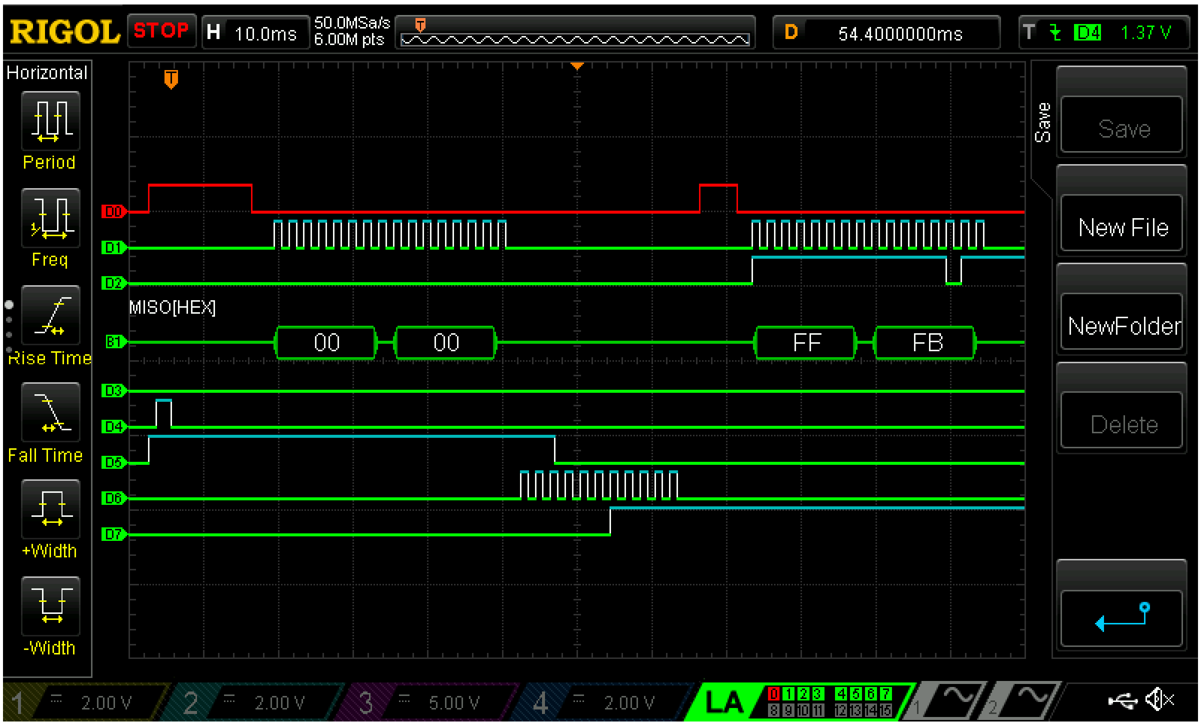

Figure 5: Functionality of "16-bit FSM with SPI Output, OVF and Initial Value's Setting Functions" OVF Event Happened

The last design which is related to 16-bit FSM with SPI output is "16-bit FSM with SPI output, zero detector and initial value's setting function".

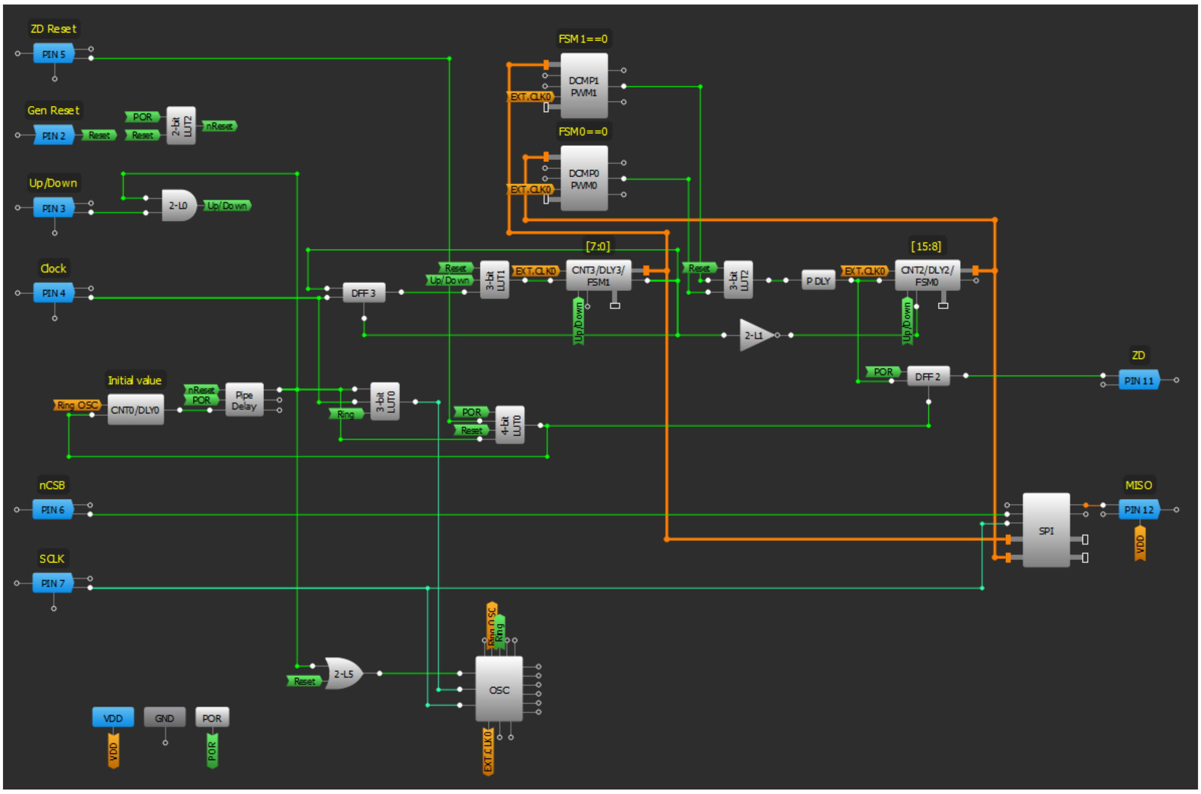

Figure 6: “16-bit FSM with SPI Output, Zero Detector and Initial Value's Setting Function”

When 16-bit counter value is equal to zero, a corresponding ZD output (Zero detector) will be set and latched. It will be reset after IC reset or when “Gen Reset” or “ZD Reset” signal arrives.

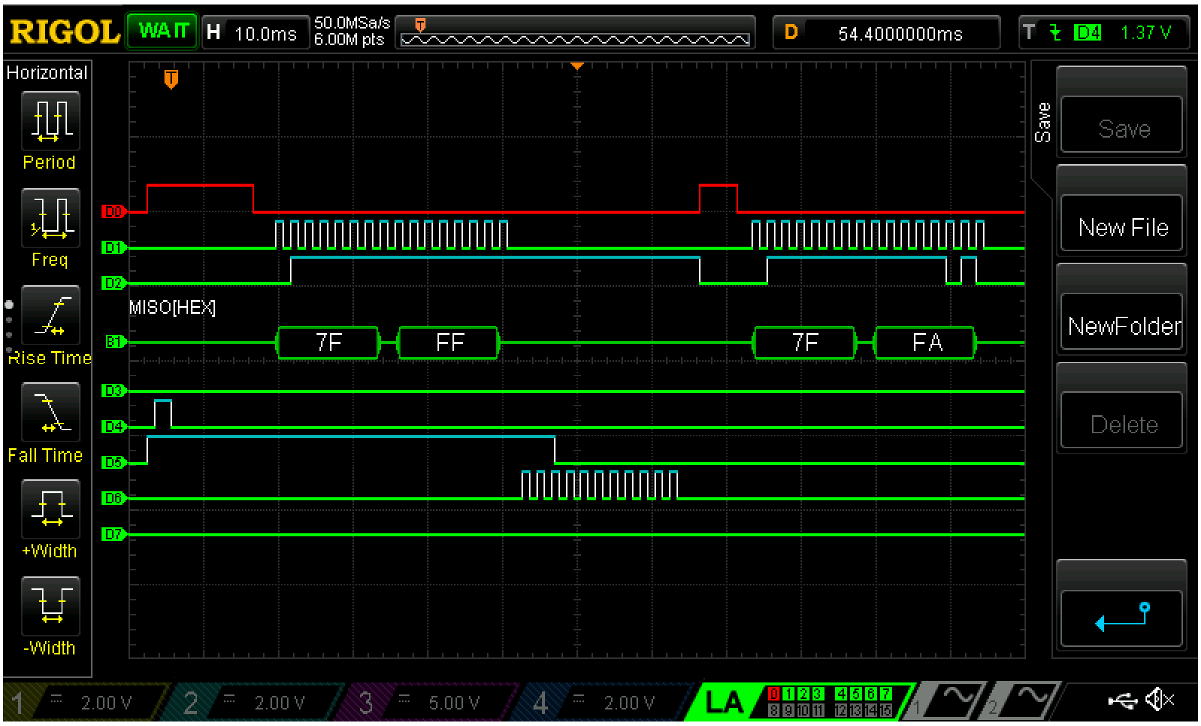

Figure 7: Functionality of “16-bit FSM with SPI Output, OVF and Initial Value’s Setting Functions”

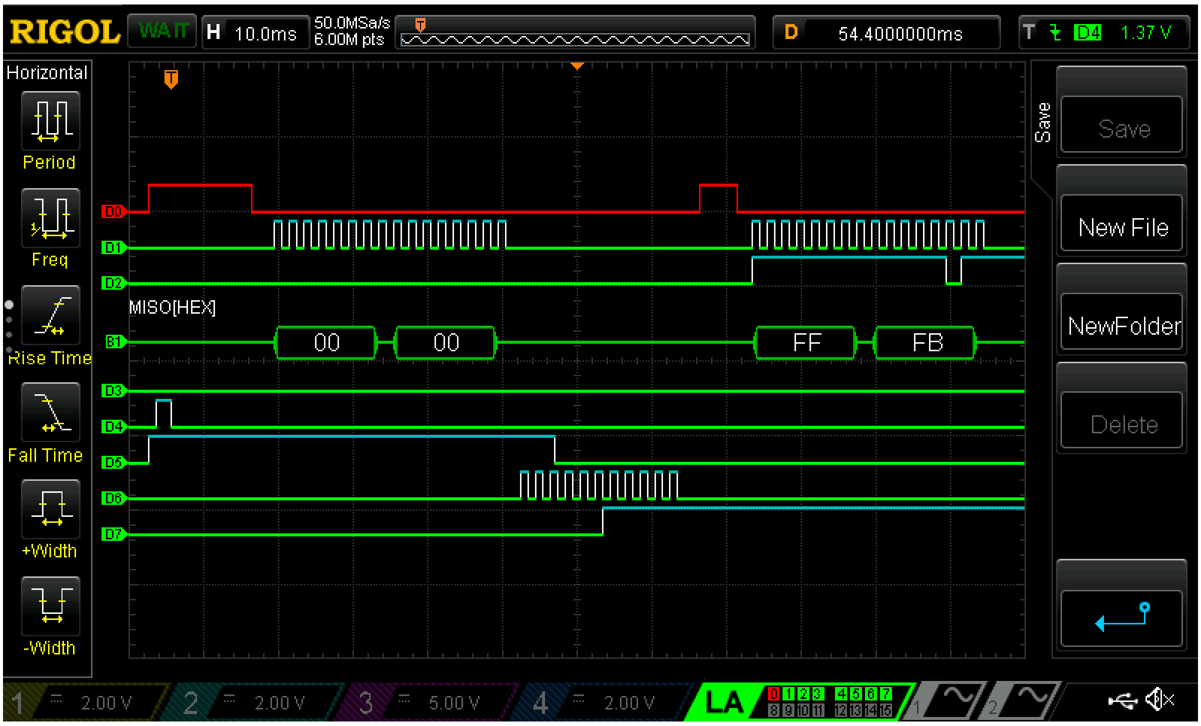

Figure 8: Functionality of "16-bit FSM with SPI Output, OVF and Initial Value's Setting Functions" ZD Event Happened

5 “16-bit FSM with I2C Output”

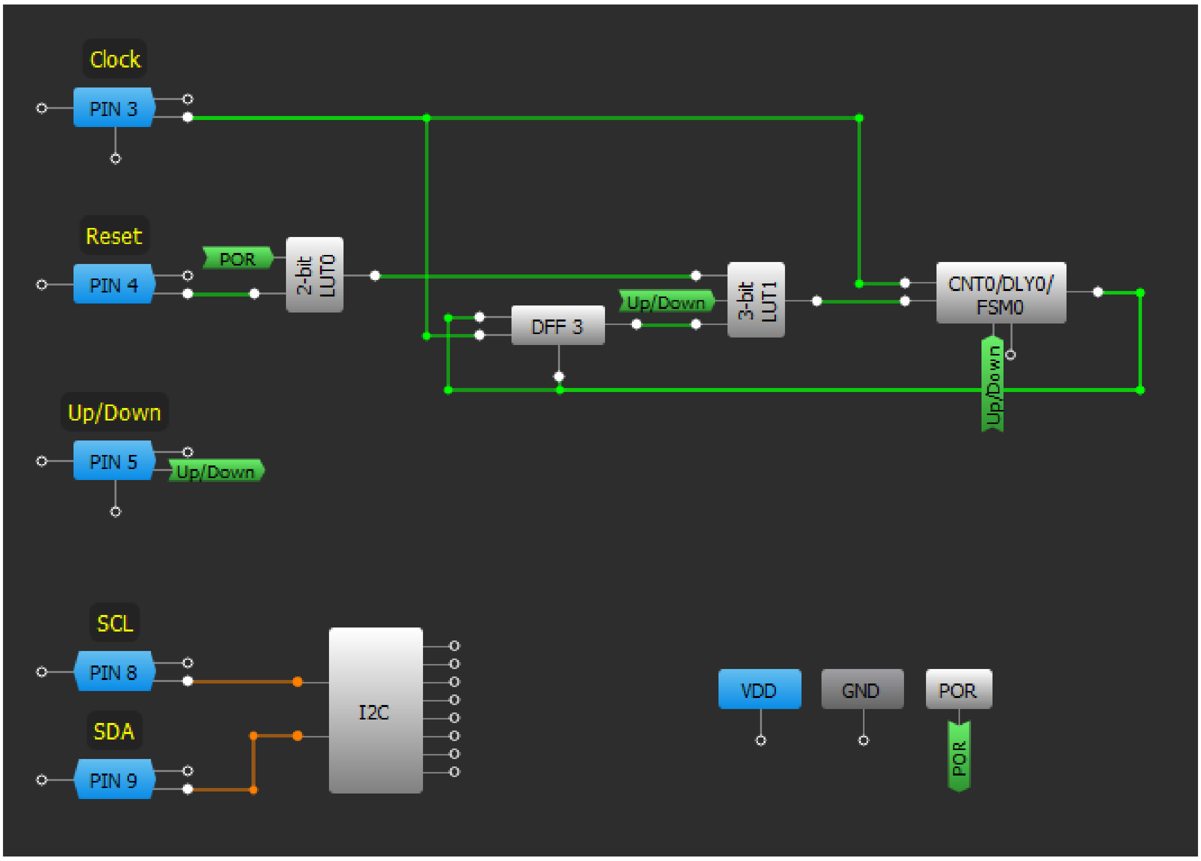

Figure 9: “16-bit FSM with I2C Output”

This design counts input clocks in an internal 16-bit register (FSM0). At any time the user can read the value via I2C, reset the 16-bit register, or change direction of counting. This design is implemented on SLG46533. There is one more FSM cell which can be used to get one 32-bit register or two independent 16-bit registers. Also, initial value can be set in the same way as in previous designs.

The DFF3 and 3-bit LUT1 have the same purpose as in “16-bit FSM with SPI output”, namely reset FSM0 by next clock when FSM0 reaches max value (0xFFFF) and Up/Down signal is HIGH, since FSM0 stops when it reaches max value and FSM0 UP input is HIGH.

D0 – PIN#3 (Clock)

D1 – PIN#4 (Reset)

D2 – PIN#5 (Up/Down)

D3 – PIN#8 (SCL)

D4 – PIN#9 (SDA)

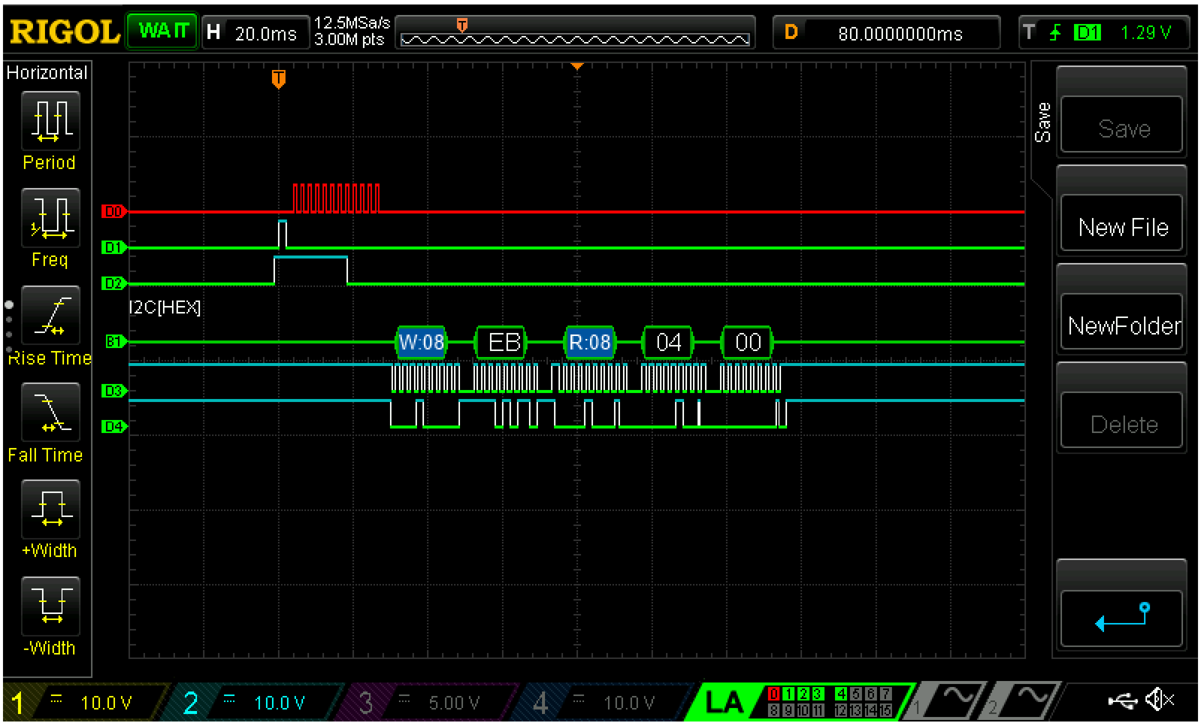

Figure 10: Functionality of “16-bit FSM with I2C Output”

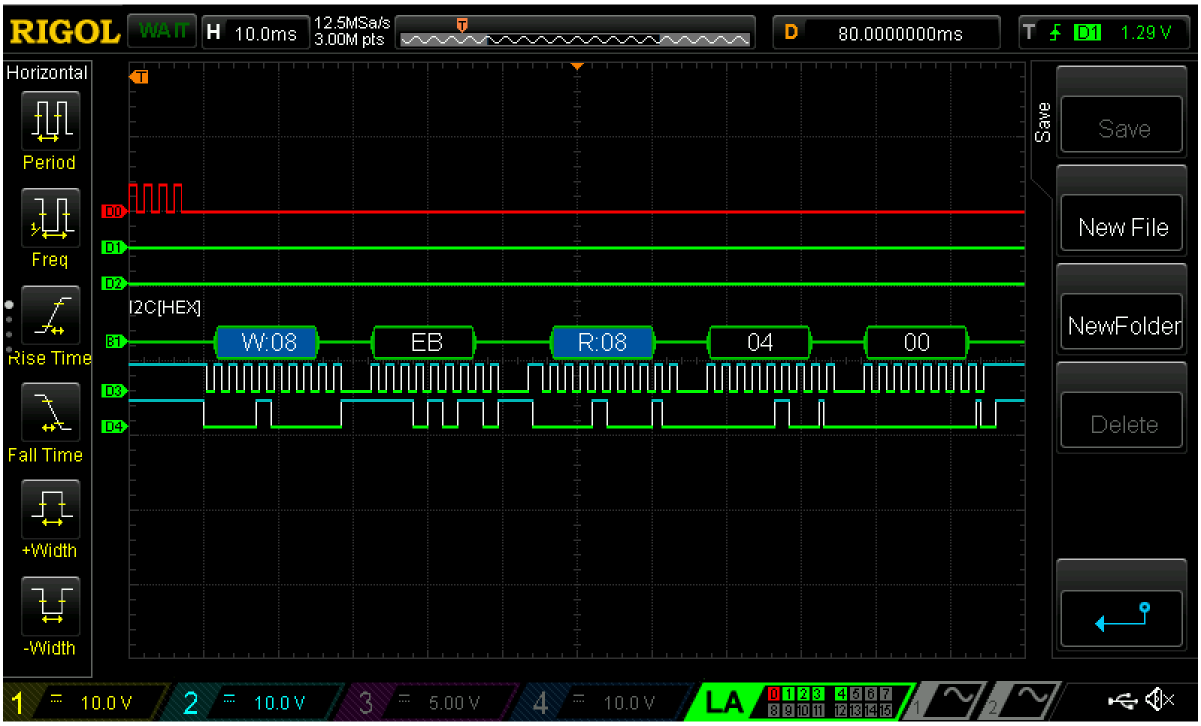

Figure 11: Zoomed Functionality of “16-bit FSM with I2C Output”

6 “9-bit FSM with Parallel Outputs”

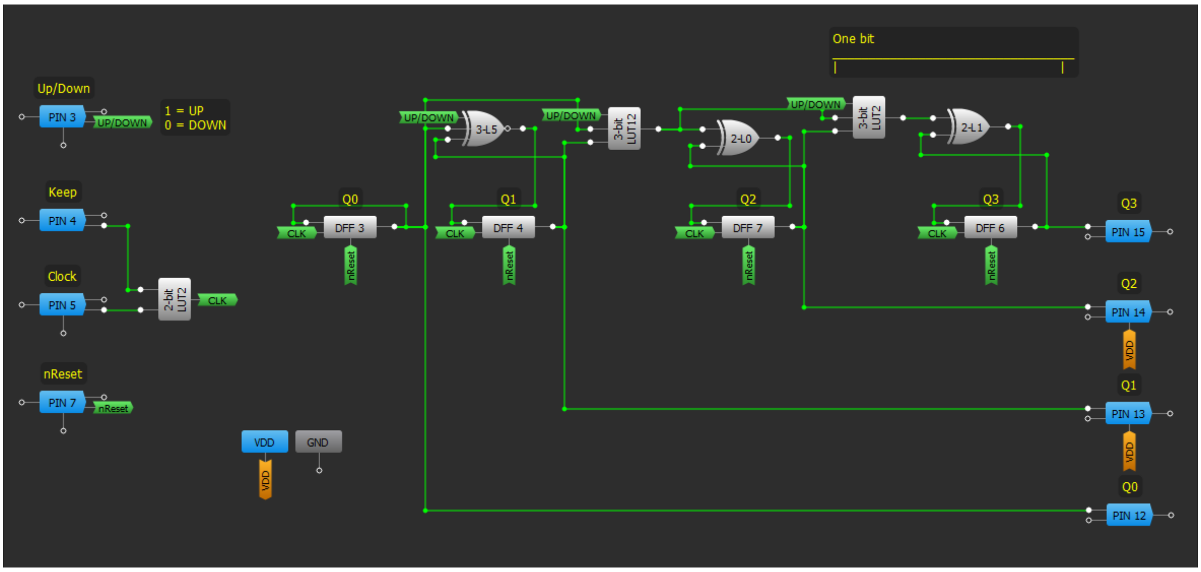

Figure 12: “4-bit FSM with Parallel Outputs”

The last design is a bidirectional counter with parallel outputs. In order to build this bidirectional counter, connected DFF in series is not enough. For building the bidirectional counter additional logic components are needed. The design is built based on DFFs and LUTs. At any time the user can reset the counter value or change direction of counting. Keep signal (active HIGH) is designated to keep counter value regardless clock signal.

D0 – PIN#3 (Up/Down)

D1 – PIN#4 (Keep)

D2 – PIN#5 (Clock)

D3 – PIN#7 (nReset)

D4 – PIN#15 (Q3)

D5 – PIN#14 (Q2)

D6 – PIN#13 (Q1)

D7 – PIN#12 (Q0)

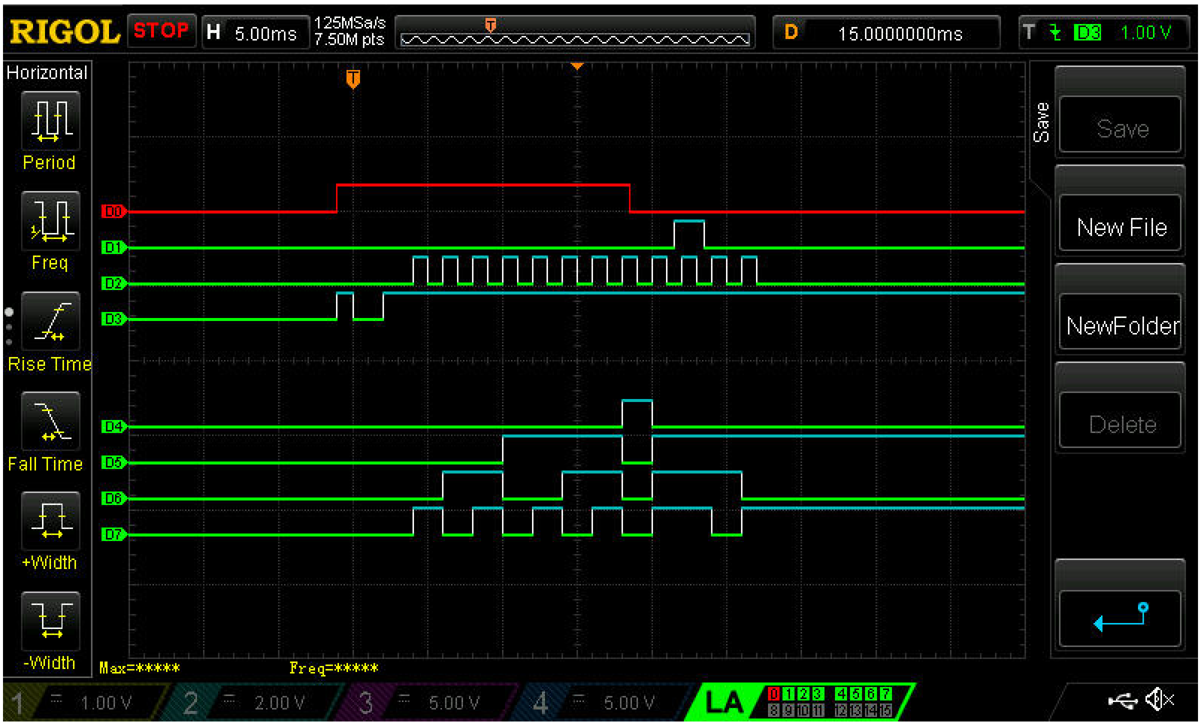

Figure 13: Functionality of “4-bit FSM with Parallel Outputs (Based on Glue Logic)”

In order to increase the number of outputs, “One bit” section (3-bit LUT2, 2-L1, DFF6) should be just copied, as is shown on Figure 14.

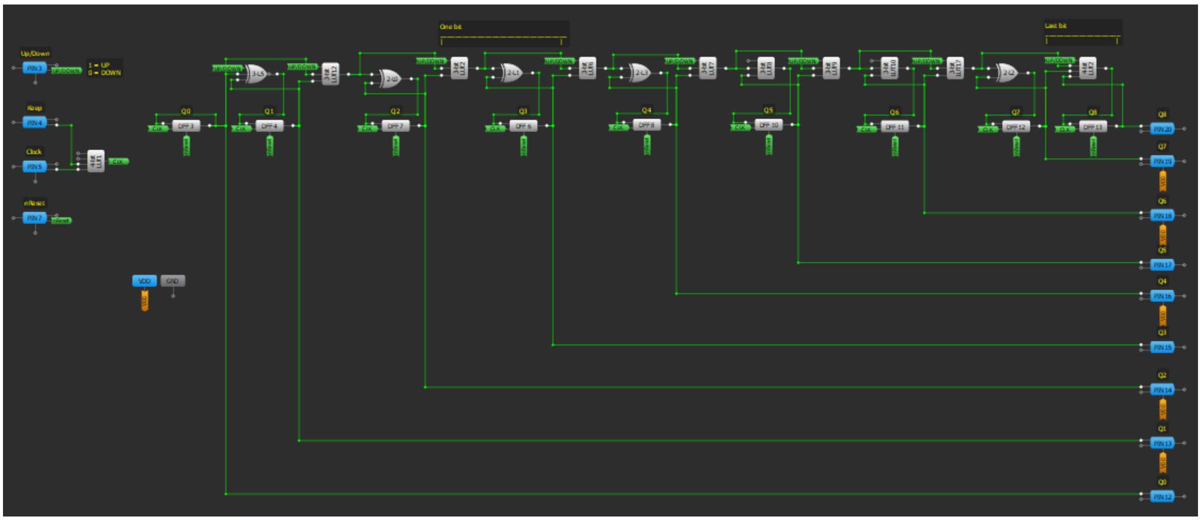

Figure 14: Functionality of “9-bit FSM with Parallel Outputs”

7 Conclusions

Counters have become an integral part a most of devices such as real time systems, speedometers, odometers, electricity meters, counting revolutions etc. GreenPAK is ideal for implementing the required counter function with lower cost and small board space.

Revision History

Revision	Date	Description
1.0	19-Jul-2019	Initial Draft

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

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/.