How to use Two_Clk_Domains
SLG47910

Abstract

ForgeFPGA has two Clock domains namely Clock_0 and Clock_1. This application shows how to design using two clock domains in a single Verilog code and design a frequency divider.

Contents

1. Terms and Definitions

FPGA	Field Programmable Gate Array
FPGA Editor	Main FPGA design and simulation window
Go Configure Software Hub	Main window for device selection
ForgeFPGA Window	Main FPGA project window for debug and IO programming
LaC	Logic-As-Clock
PLL	Phase Locked Loop
OSC	Oscillator

2. References

To find more information about ForgeFPGA™ products, please visit the website: https://www.renesas.com/eu/en/products/programmable-mixed-signal-asic-ip-products/forgefpga-low-density-fpgas. Download our free ForgeFPGA™ Designer software [1] to open the. ffpga design files [2] and view the proposed circuit design.

[1] ForgeFPGA Designer Software, Software Download and User Guide

[2 AN-013 How to use Two-Clock Domain.ffpga, ForgeFPGA Design File

[3] ForgeFPGA SLG47910, Datasheet, Renesas Electronics

3. Introduction

The FPGA Core has two main clock domains namely, Clock_0 and Clock_1. Each of these active clocks are internally connected to the sub clocks within the core.

The core is divided into four directions, on the North side of the core we have the REF_BRAM [3:0] _WRITE_CLK & REF_BRAM [3:0] _READ_CLK as outputs. Similarly, on the South side of the core we have REF_BRAM [7:4] _WRITE_CLK & REF_BRAM [7:4] _READ_CLK as outputs. On the East side of the core, we have the two LOGIC_AS_CLK0/1. This clock is an output from the FPGA Core itself. And on the West side of the FPGA Core, we have the three input clocks PLL_CLK, OSC_CLK, and GPIO through which the clock can also be supplied to the core. Using rising/falling edge of the same clock the core can utilize two clock domains CLK0 and CLK1.(See Figure 1)

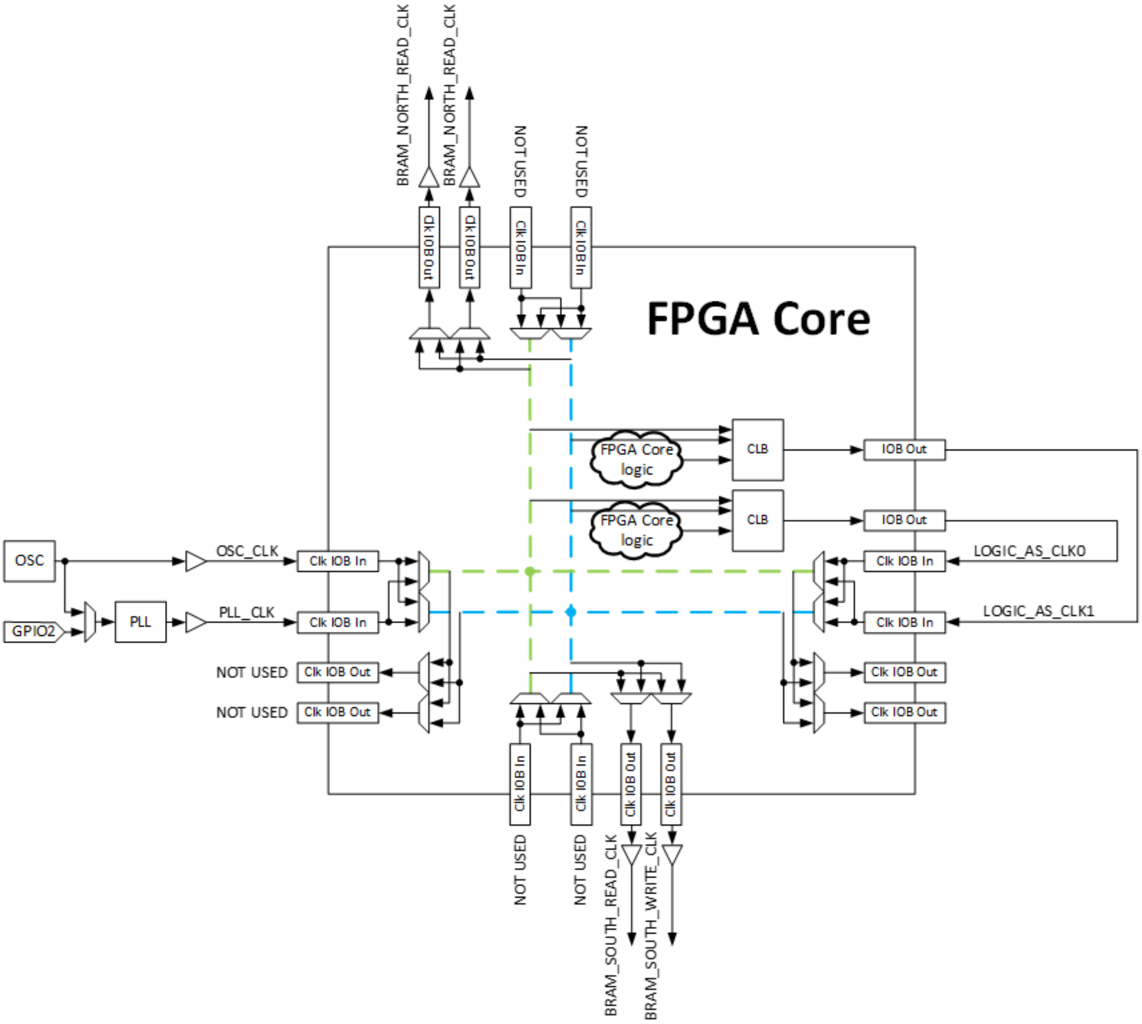


Figure 1: Clock Network

This application note is based on these clock domains and how to setup the Verilog code and the I/O Planner to work based on these two clock domains. We use the high-frequency on-chip Oscillator as our clock #1 and the PLL_CLK as our clock #2 for this application note which takes in either the oscillator clock or an external clock through GPIO2.

4. Ingredients

- ForgeFPGA Device SLG47910
- Latest Revision of ForgeFPGA Workshop software
- SLG47910 Development Board and Adaptor Board

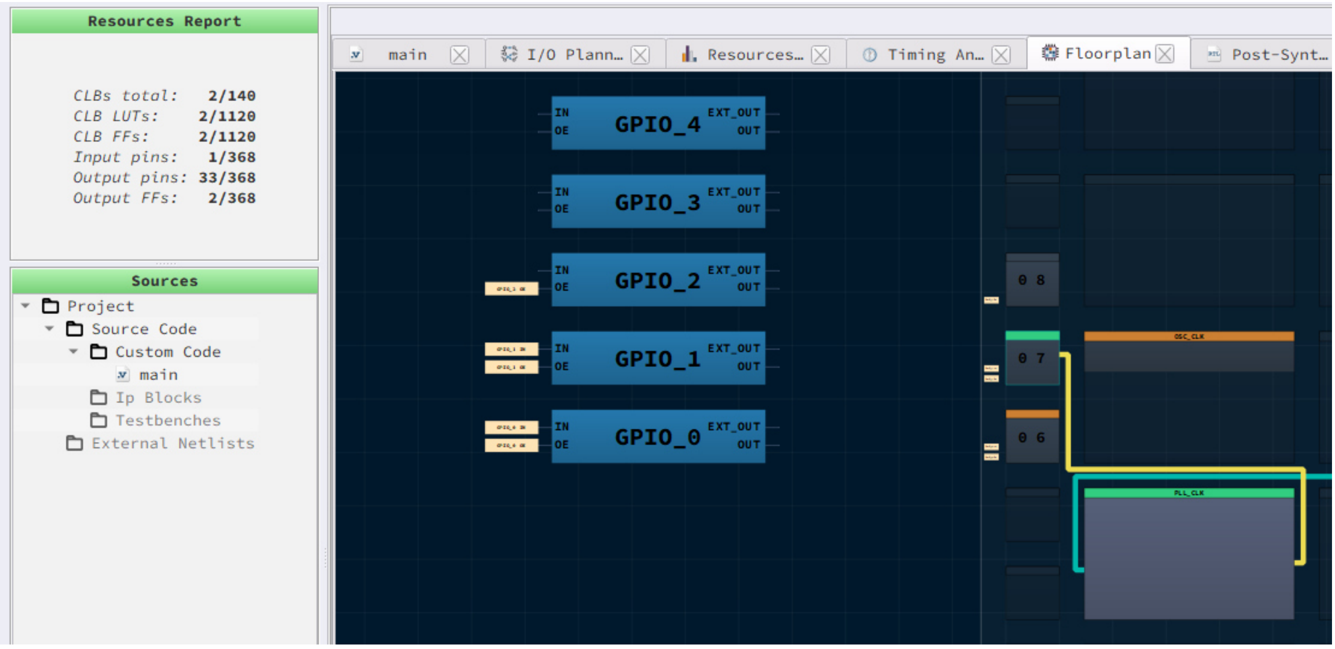
5. Verilog Code

Below is the Verilog Code on how to use two clocks: PLL_CLK and OSC_CLK in this application note. User can find the complete code [2]. Using the below Verilog code, the user can design a Frequency Divider. The Frequency Divider is implemented using a FF using the osc_clk and pll_clk.

```verilog
(* top  *) module two_clk_domain (
  (* iopad_external_pin, clkbuf_inhibit *) input pll_clk,  // clock_0
  (* iopad_external_pin, clkbuf_inhibit *) input osc_clk,   // clock_1
  (* iopad_external_pin *) input rst,
  (* iopad_external_pin *) output ext_clk_oe,//to enable GPIO2 to function as input
  (* iopad_external_pin *) output OSC_CTRL_EN,
  // PLL Settings
  (* iopad_external_pin *) output PLL_EN,
  (* iopad_external_pin *) output PLL_SELECTION,
  (* iopad_external_pin *) output PLL_BYPASS,
  (* iopad_external_pin *) output [5:0] PLL_REFDIV,
  (* iopad_external_pin *) output [11:0] PLL_FBDIV,
  (* iopad_external_pin *) output [2:0] PLL_POSTDIV1, PLL_POSTDIV2,
  (* iopad_external_pin *) output reg osc_div,
  (* iopad_external_pin *) output osc_div_oe,
  (* iopad_external_pin *) output reg pll_div,
  (* iopad_external_pin *) output pll_div_oe
);

// divider by 2 frequency divider
  always @(posedge osc_clk) begin
    if (!rst) begin
      osc_div <= 1'b0;
    end else begin
      osc_div <= ~osc_div;
    end
  end

// divider by 2 frequency divider
  always @(posedge pll_clk) begin
    if (!rst) begin
      pll_div <= 1'b0;
    end else begin
      pll_div <= ~pll_div;
    end
  end

// if 1'b0, then set GPIO as input
  assign ext_clk_oe = 1'b0;
// if 1'b1, then set GPIO as output
  assign osc_div_oe = 1'b1;
// if 1'b1, then set GPIO as output
  assign pll_div_oe = 1'b1;
// if 1'b1, then OSC is enabled
```

```
  assign OSC_CTRL_EN = 1'b1;
// if 1'b1, then PLL is enabled
  assign PLL_EN = 1'b1;
// if 1'b1, then input clock goes directly without any divider
  assign PLL_BYPASS = 1'b0;
// if 1'b1, then input clock takes from GPIO2, otherwise takes from OSC
  assign PLL_SELECTION = 1'b1;
// sets the reference divide value from 1 to 63
  assign PLL_REFDIV = 6'd1;
// sets the PLL Feedback Divide value from 16 to 400
  assign PLL_FBDIV = 12'd55;
// Sets the PLL Output Divider1 value from 1 to 7
  assign PLL_POSTDIV1 = 3'd7;
// Sets the PLL Output Divider2 value from 1 to 7
  assign PLL_POSTDIV2 = 3'd7;
endmodule
```

6. Floorplan: CLB Utilization



Resources Report

```
CLBs total:    2/140
CLB LUTs:     2/1120
CLB FFs:      2/1120
Input pins:    1/368
Output pins:  33/368
Output FFs:    2/368
```

Figure 2: CLB Utilization

7. Design Steps

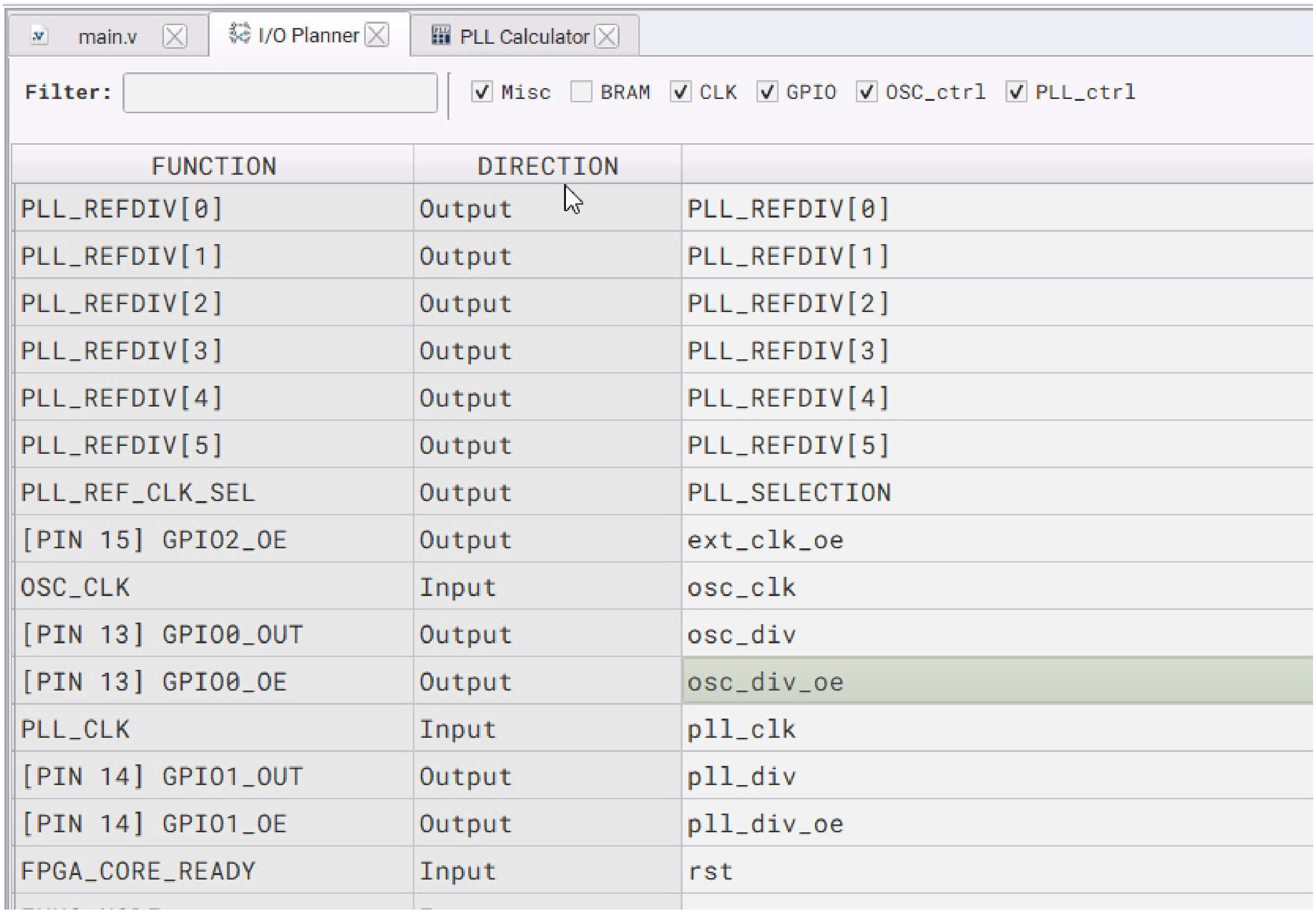
1. Launch the latest version of the Go Configure Software Hub. Select the SLG47910V device and the ForgeFPGA Workshop software will load.

2. Download the design example AN-FG-013 How to use two clock domain.ffpga. If you are not familiar with the ForgeFPGA Workshop software, review the Four-Bit Counter application notes that covers the basic design steps.

3. Open the AN-FG-013 How to use two clock domain.ffpga file after downloading.

4. Open the FPGA editor and review the Verilog code. There is a main code with the module name two_clk_domain,which is the top module defining the whole design. This is a Frequency Divider code using DFF.

5. Open the IO planner tab on the FPGA editor and review the pin assignment. The Pin assignment in the IO Planner is what makes the Two Clock connection work as expected. (Figure 2)



Filter: ☑ Misc □ BRAM ☑ CLK ☑ GPIO ☑ OSC_ctrl ☑ PLL_ctrl

FUNCTION	DIRECTION	
PLL_REFDIV[0]	Output	PLL_REFDIV[0]
PLL_REFDIV[1]	Output	PLL_REFDIV[1]
PLL_REFDIV[2]	Output	PLL_REFDIV[2]
PLL_REFDIV[3]	Output	PLL_REFDIV[3]
PLL_REFDIV[4]	Output	PLL_REFDIV[4]
PLL_REFDIV[5]	Output	PLL_REFDIV[5]
PLL_REF_CLK_SEL	Output	PLL_SELECTION
[PIN 15] GPIO2_OE	Output	ext_clk_oe
OSC_CLK	Input	osc_clk
[PIN 13] GPIO0_OUT	Output	osc_div
[PIN 13] GPIO0_OE	Output	osc_div_oe
PLL_CLK	Input	pll_clk
[PIN 14] GPIO1_OUT	Output	pll_div
[PIN 14] GPIO1_OE	Output	pll_div_oe
FPGA_CORE_READY	Input	rst

Figure 2: IO Planner

6. Next select the Synthesize button on the lower left side of the FPGA editor. Select the Generate Bitstream button on the lower left side of the FPGA editor. Check the Logger and Issues tabs to make sure that the bit steam was generated correctly.

7. Now click on the Floorplan tab and see the CLB utilization (Figure 2). Press the Ctrl and the mouse wheel to zoom-in.

8. Connect the Development Board and attach it to Adaptor Board with the SLG47910 part in the socket on it. Click on the Debug button on the ForgeFPGA Workshop studio and select Emulation (Figure 3).

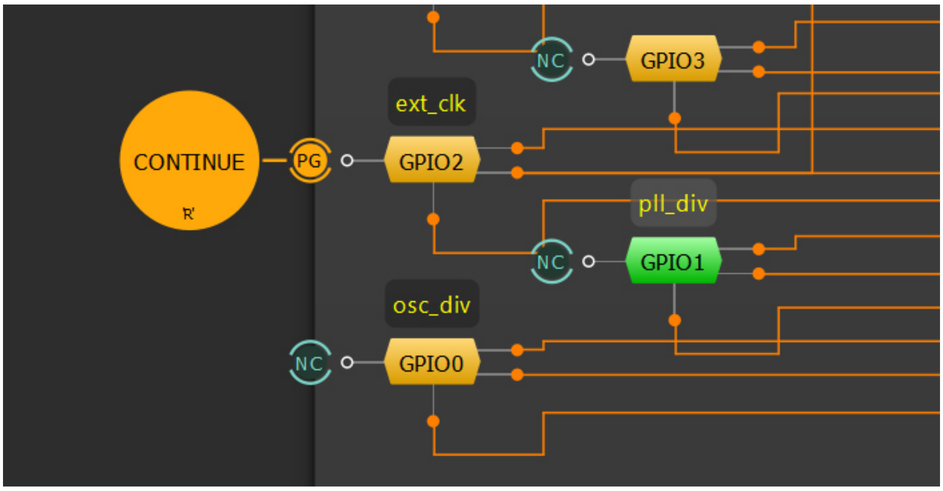


Figure 3: GPIO Connection

9. Connect GPIO2 to Synchronous Logic Generator and set a pattern of choice in the Generator. The input frequency will be divided by two due to flip-flop connected to the output and can be observed on GPIO0 and GPIO1 depending on the settings. It can be observed that GPIO2 is set at 20MHz input which results in 10MHz output at GPIO1 and GPIO2 at 25MHz frequency as the OSC runs at 50MHz (See Figure 5)

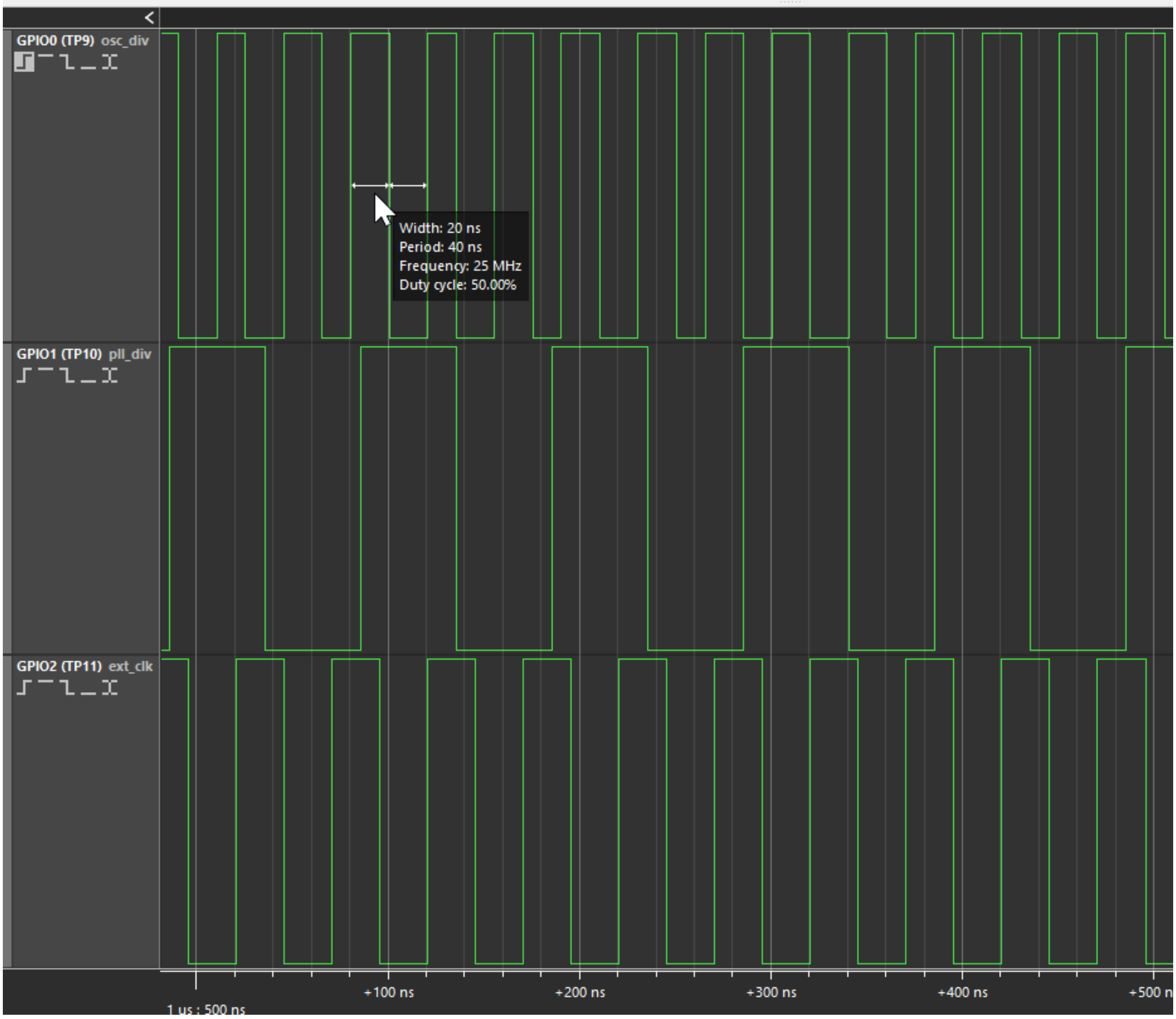


Figure 4: Waveform using External Clock

8. Conclusion

This application note shows how the Frequency Divider is designed using two different clock domains and how the input-output ports are assigned in IO Planner. This procedure can be utilized for any design. This testcase is available for download (How to use two clock domains.ffpga). If interested, please contact the ForgeFPGA Business Support Team.

9. Revision History

Revision	Date	Description
1.00	May 03,2024	Initial release.
2.0	July 17, 2024	Updated as per ForgeFPGA Workshop v6.43

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/.