How to use the Logic-As-Clock SLG47910

Abstract

This application shows how to connect the FPGA Board to run using an external clock source though GPIO. This application note comes complete with design files which can be found in the References section. This Application note is written according to ForgeFPGA Workshop v6.43

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

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-FG-009 How to use Logic-As-Clock.ffpga, ForgeFPGA Design File

[3] SLG47910, Preliminary Datasheet, Renesas Electronics

3. Introduction

The SLG47910 Development board has the capability to work with 3 types of Global Clocks Trees: PLL Clock, Oscillator Clock and Logic-As-Clock 0/1(LaC). This application note focuses on how to setup the ForgeFPGA Workshop software to operate on External Logic.

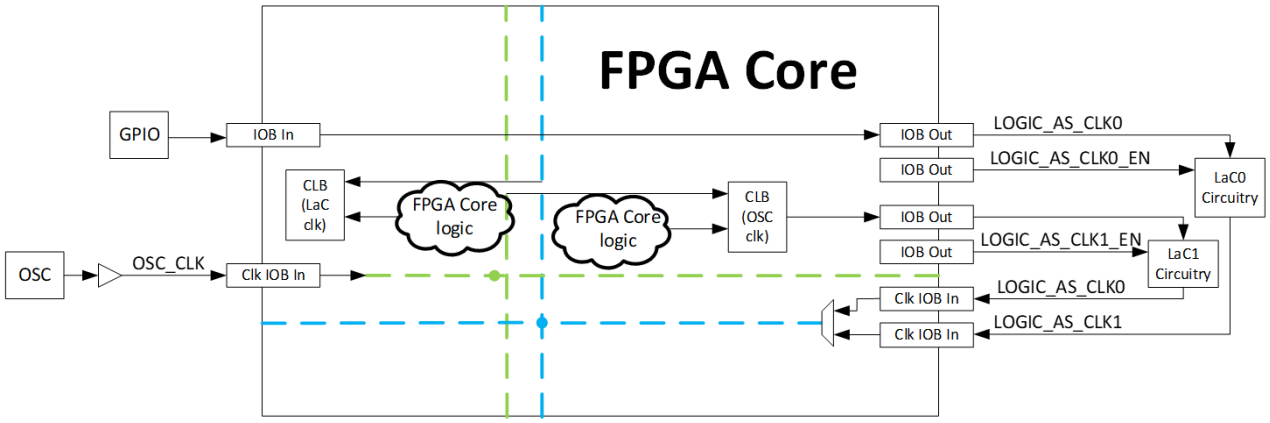

Figure 1 : LaC Circuitc

We will be using the example of a simple Frequency Divider to showcase the connections for External Logic. The input & output ports of the design must be assigned to the IOB Titles in the IO Planner of the ForgeFPGA Software according to Figure 1.

To enable the LaC functionality of the software, the user needs to assign a 1'b1 to the signal: LOGIC_AS_CLK0_EN in the IO Planner. For this purpose, user signal which is to be used as clock should first be output through Logic-IOB (REF_LOGIC_AS_CLK0/1) and then looped back into core as clock through clock-IOB (LOGIC_AS_CLK0/1)

4. Ingredients

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

5. Verilog Code

Shown below is the (*top*) module called logic_as_clk. It is available for download [AN-FG-009 How to use Logic-As-Clock.ffpga].

```
(* top *) module logic_as_clk (
  (* iopad_external_pin, clkbuf_inhibit *) input clk,
  (* iopad_external_pin *) input nreset,
  (* iopad_external_pin *) input clk_in,//GPIO0
  (* iopad_external_pin *) output clk_out,
  (* iopad_external_pin *) output out,//GPIO3
  (* iopad_external_pin *) output out_oe,
  (* iopad_external_pin *) output clk_en
);

  reg dff;
```

```
  reg nrst;

 //oe
  assign out_oe = 1;
 //enable loopback clock functionality
  assign clk_en = 1;

  assign clk_out = clk_in;
  assign out = dff;

  //Synchronize nReset
  always @(posedge clk) begin
    nrst <= nreset;
  end

  always @(posedge clk) begin
    if(!nrst)
      dff <= 1'b1;
    else
      dff <= ~dff;
  end

endmodule
```

6. Floorplan: CLB Utilization

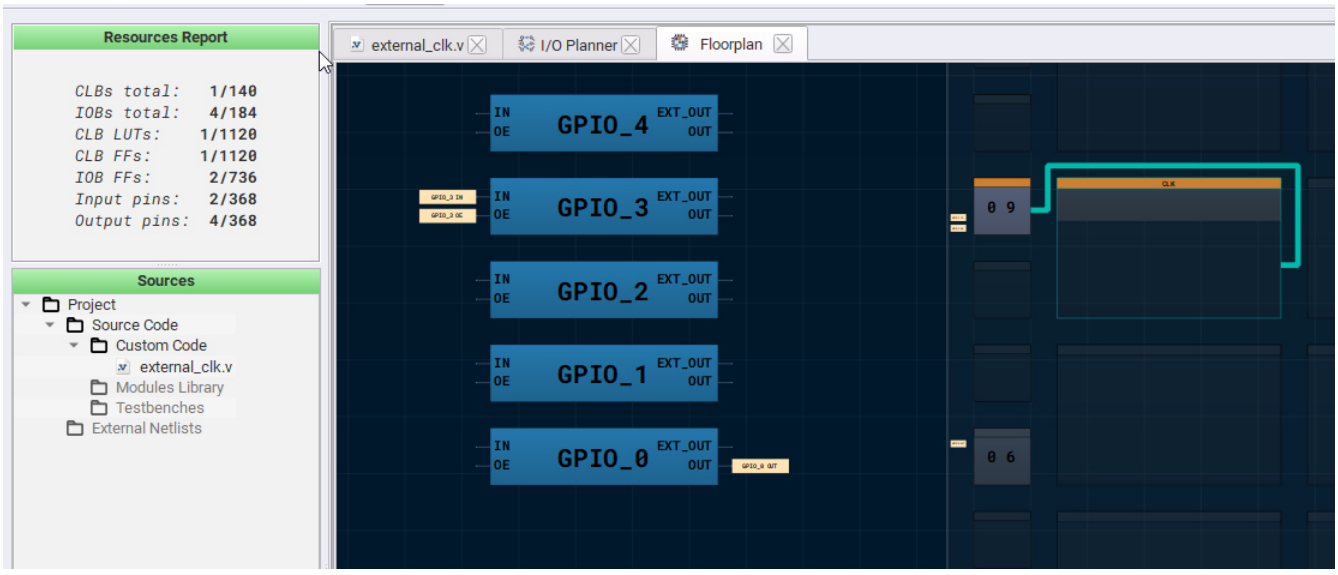

Figure 1: Floorplan & CLB Utilization

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-009 How to use Logic-As-Clock.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-009 How to use Logic-As-Clock.ffpga file after downloading.

4. Open the FPGA editor and review the Verilog code. There is a main code with the module name logic_as_clk,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 External Clock connection work as expected. (Figure 2)

FUNCTION	DIRECTION	
LOGIC_AS_CLK0	Input	clk
LOGIC_AS_CLK0_EN	Output	clk_en
[PIN 13] GPIO0_IN	Input	clk_in
REF_LOGIC_AS_CLK0	Output	clk_out
FPGA_CORE_READY	Input	nreset
[PIN 16] GPIO3_OUT	Output	out
[PIN 16] GPIO3_OE	Output	out_oe
LOGIC_AS_CLK1	Input	

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).

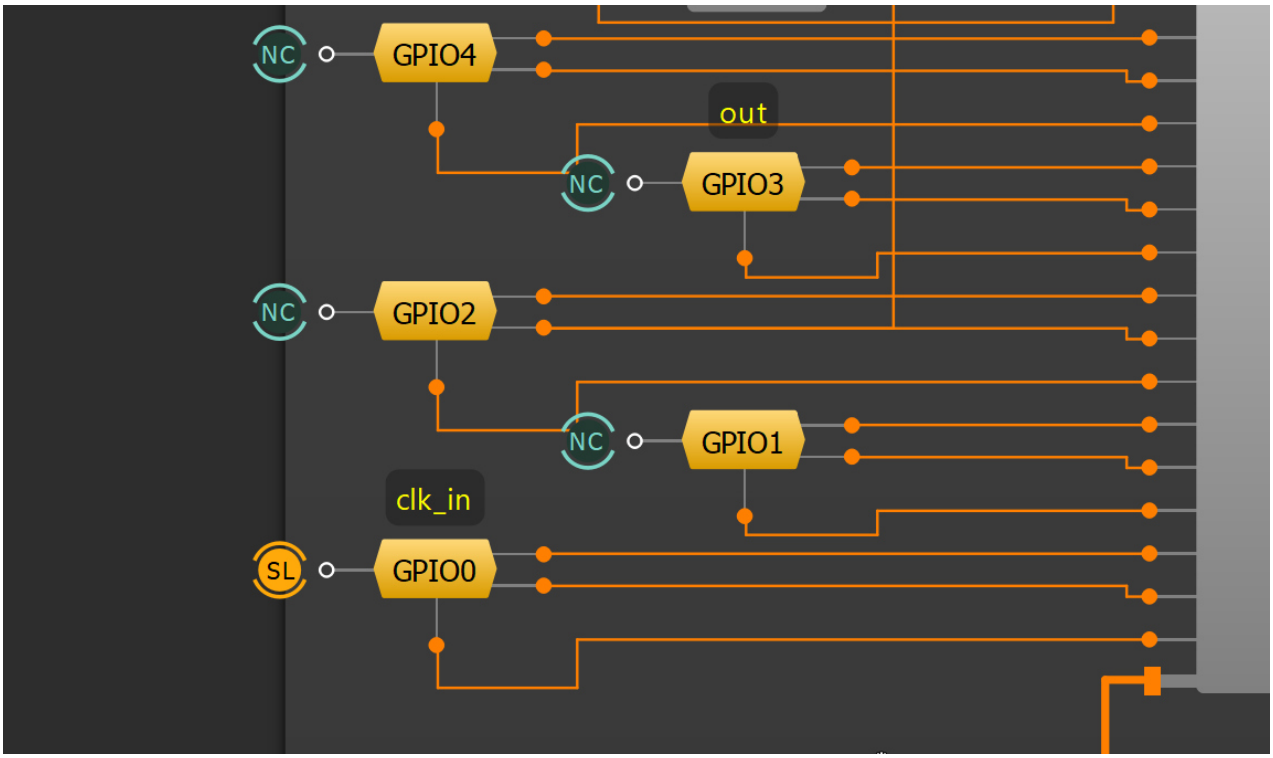

Figure 3: GPIO Connection

9. Connect the GPIO0(clk_in) to the Synchronous Logic Generator and produce 1 MHz clock frequency on it and observe the output from GPIO3 on built-in Logic Analyzer(Figure 4). The input frequency will be divided by two due to flip-flop connected to the output.

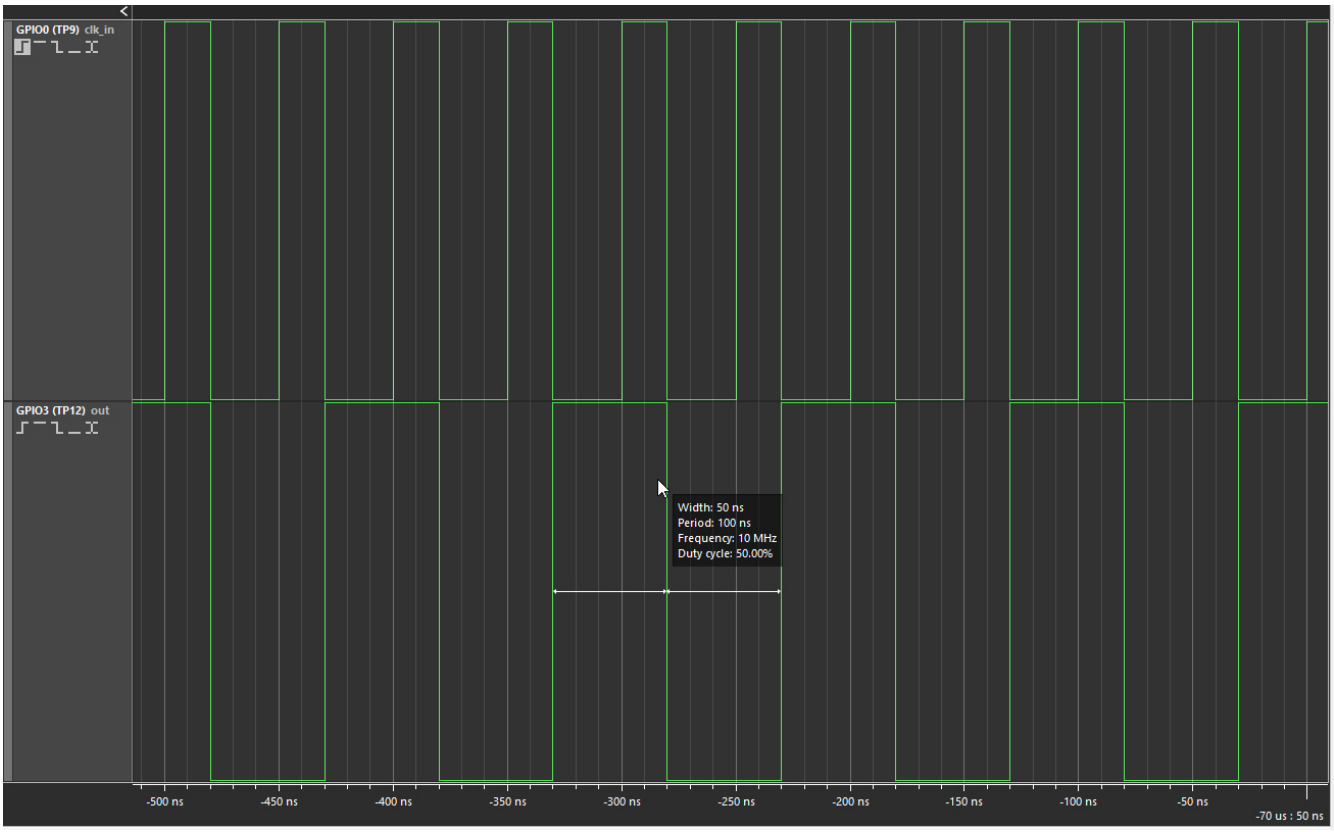

Figure 4: Waveform using External Clock

8. Conclusion

This application note shows how the Frequency Divider is designed using a Logic_As_Clock 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 (AN-FG-009 How to use Logic_As_Clock.ffpga). If interested, please contact the ForgeFPGA Business Support Team.

9. Revision History

Revision	Date	Description
1.00	Jul 22, 2022	Initial release.
2.0	Feb 23,2024	Updated according to BB revision
3.0	July 17,2024	Updated according to Software 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

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.