Application Note

SLG47910V
How to Drive PLL from Oscillator

Abstract

This application shows how to drive the PLL either from the Oscillator or from the external clk (GPIO2) using the SLG47910 ForgeFPGA and Development Board. This application note comes complete with design files which can be found in the Reference Section.

Contents

1. Terms and Definitions

CLB	Configuration Logic Block
HDL Editor	Workspace where Verilog code is entered
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
PLL	Phase Locked Loop
OSC	Oscillator

2. Reference

For related documents and software, please visit: 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 files [2] and view the proposed circuit design.

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

[2] AN-FG-006 How to drive PLL from Oscillator.ffpga, ForgeFPGA Design File

[3] SLG47910, Preliminary Datasheet, Renesas Semiconductor

3. Introduction

3.1 PLL connections

The SLG47910 includes a low-power, wide input, and output Phase-Locked Loop (PLL) for use in applications requiring various frequencies.

The Phase Locked Loop (PLL) can be operated via two clock sources, the internal frequency from the oscillator (50 MHz) or the external clock, routed through GPIO2 pin. The PLL_REF_CLK_SEL input signal is used to select between the two clock sources for PLL.

When the PLL_REF_CLK_SEL signal is LOW, then the clock input to PLL is from the 50MHz OSC via REF_CLK_OSC. When the PLL_REF_CLK_SEL signal is HIGH, then the clock input to the PLL is from an external clock from GPIO 2 via REF_CLK_EXT.

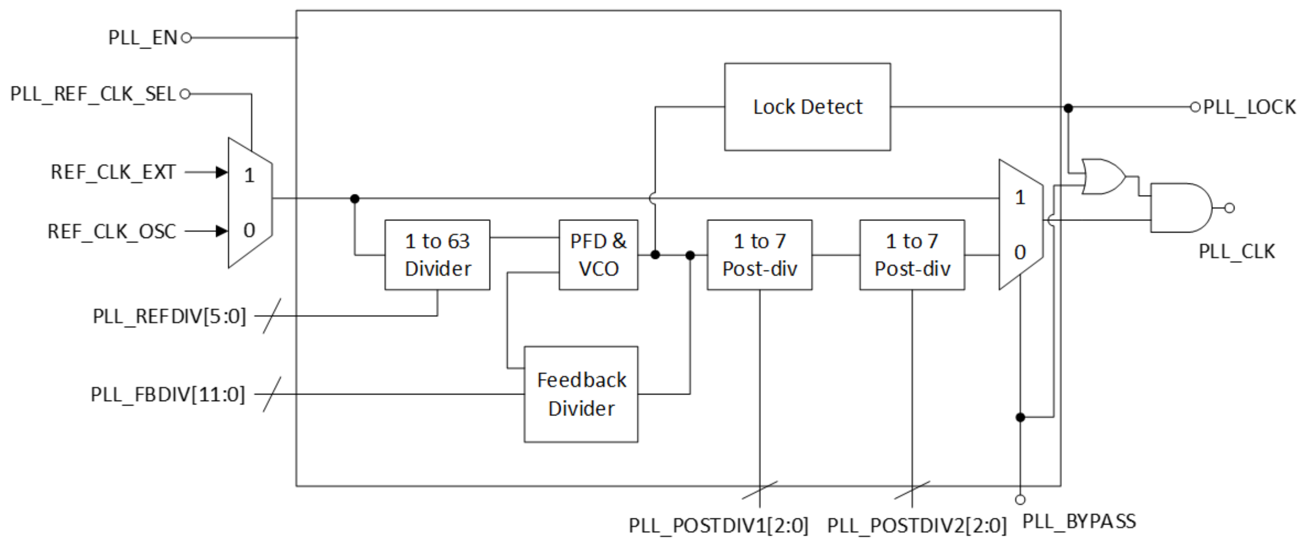


Figure 1: System Design

3.2 Signal Description

3.2.1. Inputs - Clock & Control Inputs

Clock and control inputs determine the input clock source and intended output frequency.

PLL_REF_CLK_SEL – Selects the PLL Input Clock source between the internal 50 MHz OSC or an external clock, originating from GPIO2. When LOW the clock input to the PLL is the OSC.

PLL_BYP – BYP is an active HIGH signal that asserts a direct path between the clock input and PLL_CLK.

PLL_REFDIV [5:0] – Sets the reference divide value from 1 to 63.

PLL_FBDIV [11:0] – Sets the PLL Feedback Divide value from 16 to 400.

PLL_POSTDIV1[2:0] & PLL_POSTDIV2[2:0] – The two stages of post-dividers are used to divide down the VCO Frequency before the PLL_FOUT clock output. Each Post-divider has options for division from 1 to 7. Total post divide is POSTDIV1*POSTDIV2.

3.2.2. Clock & Control Outputs

PLL_CLK – PLL output clock (Depending on the Lock state of PLL)

PLL_LOCK – Indicates the number of cycle slips between the feedback clock and the Phase Frequency Detector for 256 consecutive cycles.

PLL_EN (Input) – Power-on for PLL. Active HIGH.

The behavior of the SLG47910's PLL is to receive a reference frequency and either divide or multiply the frequency value per the following equation, where $f_{reference\_clock}$ is the reference frequency of the external clock source (GPIO2) or 50 MHz OSC, chosen through PLL_REF_CLK_SEL:

$$\text{FOUT} = \frac{f_{reference}\ X\ FBDIV}{REFDIV\ X\ POSTDIV1\ X\ POSTDIV2}$$

Using larger values for the variables in the numerator and denominator will reduce clock jitter at the expense of increased current consumption. Below is the table specifying the different values of FBDIV, REFDIV & POSTDIV when $f_{reference}$ is set to 50MHz

Table 1: PLL FOUT Values

FREF (MHz)	REFDIV	FBDIV	POSTDIV1	POSTDIV2	PLL_FOUT (MHz)
50	2	32	5	4	40
50	3	40	7	7	13.6
50	1	16	4	2	100

The FOUT values will be correct and achievable only for valid values of FBDIV, REFDIV and POSTDIV values.

The PLL_BYP Signal is an active HI signal. When the PLL_BYP signal is set as 1, then the PLL_CLK signal runs at the frequency directly from either REF_CLK_EXT or REF_CLK_OSC. However, when the PLL_BYP is set as 0, then the PLL_CLK value is assigned according to the Equation above and Table 1.

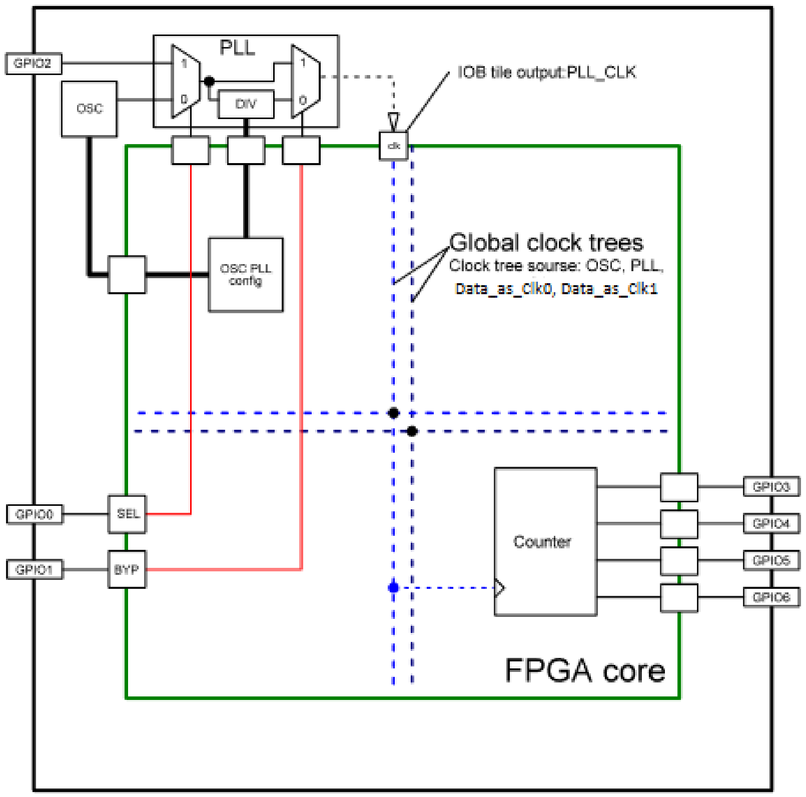


Figure 2: Block diagram of the design

4. Ingredients

- SLG47910 Device
- FPGADEVPAK Development Board with USB cable and power supply
- FPGAPAK Socket Adaptor Board
- Latest Revision of ForgeFPGA Workshop software

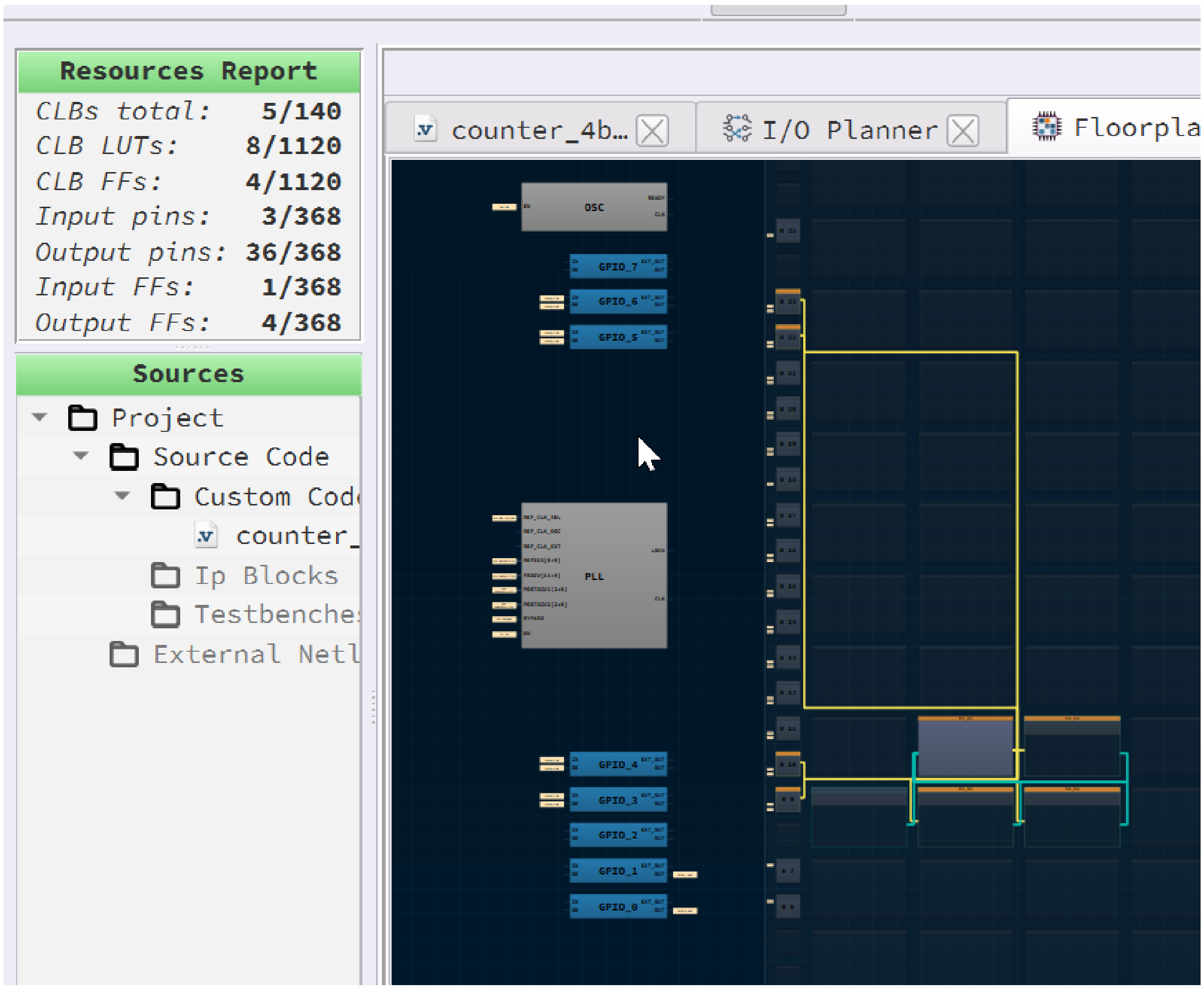
5. Verilog Code

Shown below is the (*top*) module named counter_4bit_pll. The Verilog code for How to drive PLL can be found in the complete design example. It is available for download (AN-FG-006 Drive PLL from Oscillator.ffpga)

```
// AN-003 How to drive PLL from OSC

(* top *) module counter_4bit_pll (
  //CLK
  (* iopad_external_pin, clkbuf_inhibit *) input pll_clk,
  //POR
  (* iopad_external_pin *) input nreset,
  //OSC
  (* iopad_external_pin *) output OSC_CTRL_EN,
  //PLL
  (* iopad_external_pin *) input sel,//GPIO0
  (* iopad_external_pin *) input byp,//GPIO1
  // Selects the PLL Input Clock source between the internal 50MHz OSC and an
external clock.
  (* iopad_external_pin *) output PLL_REF_CLK_SELECTION,
  // BYP is an active HI signal that asserts a direct path between the clock input
and FOUT.
```

```
  (* iopad_external_pin *) output PLL_CTRL_BYPASS,PLL_CTRL_EN,
  // Sets the reference divide value from 1 to 63.
  (* iopad_external_pin *) output [5:0] PLL_CTRL_REFDIV,
  // Sets the PLL Feedback Divide value from 16 to 400.
  (* iopad_external_pin *) output [11:0] PLL_CTRL_FBDIV,
  // Sets the PLL Output Dividers values from 1 to 7.
  (* iopad_external_pin *) output [2:0] PLL_CTRL_POSTDIV1, PLL_CTRL_POSTDIV2,
  (* iopad_external_pin *) output reg [3:0] counter ,
  (* iopad_external_pin *) output counter_oe0,
  (* iopad_external_pin *) output counter_oe1,
  (* iopad_external_pin *) output counter_oe2,
  (* iopad_external_pin *) output counter_oe3
);

 //OE's settings
 assign counter_oe0 = 1'b1;
 assign counter_oe1 = 1'b1;
 assign counter_oe2 = 1'b1;
 assign counter_oe3 = 1'b1;

 //OSC settings
 assign OSC_CTRL_EN = 1'b1; // Oscillator operates at 50MHz

 //PLL settings
 assign PLL_CTRL_EN = 1;
 assign PLL_CTRL_BYPASS = byp;
 assign PLL_REF_CLK_SELECTION = sel;
 assign PLL_CTRL_REFDIV = 1;
 assign PLL_CTRL_FBDIV = 20;
 assign PLL_CTRL_POSTDIV1 = 5;
 assign PLL_CTRL_POSTDIV2 = 2;

 reg nrst;

 always @(posedge pll_clk) begin
   nrst <= nreset;
 end

 //Counter
 always @(posedge pll_clk) begin
   if (!nrst)
     counter <= 4'h0;
   else
     counter <= counter + 4'h1;
  end

endmodule
```

6. Floorplan: CLB Utilization



Resources Report	
CLBs total:	5/140
CLB LUTs:	8/1120
CLB FFs:	4/1120
Input pins:	3/368
Output pins:	36/368
Input FFs:	1/368
Output FFs:	4/368

Sources

- Project
  - Source Code
    - Custom Cod
      - counter_
    - Ip Blocks
    - Testbenche
  - External Netl

Figure 3: Floorplan

The Floorplan tab in the FPGA Editor shows the placement of the CLBs, FFs and their connections to IOB blocks.

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-006 Drive PLL from Oscillator.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-006 Drive PLL from Oscillator.ffpga file after downloading.

4. Open the FPGA editor and review the Verilog code. There is a main code with the module name counter_4bit_pll which is the top module defining the whole design.

5. Open the IO planner tab on the FPGA editor and review the pin assignment.

6. Next select the Synthesize button on the lower left side of the FPGA editor.

7. 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.

8. Now click on the Floorplan tab and see the CLB utilization (Figure 3). Press the Ctrl and the mouse wheel to zoom-in. Confirm that the IOs selected in the IO Planner are shown in the floorplan

9. 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 (Figure 5) and select Emulation.

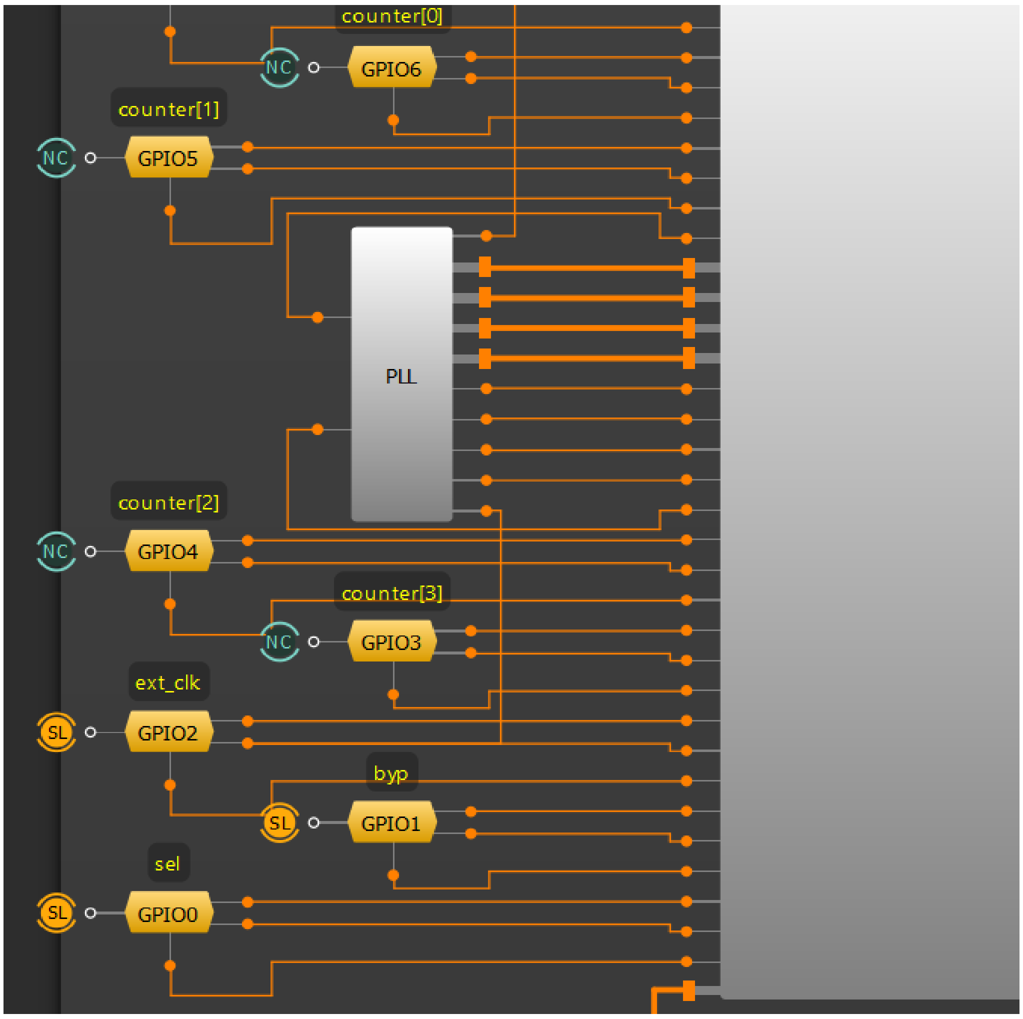


Figure 5: GPIO Connections to generate waveform during Emulation

10. Connect the GPIO2 to the Synchronous Logic Generator, Configure GPIO0 and GPIO1 as buttons or synchronous Logic Generators to function as sel and byp signals for PLL_REF_CTRL_SEL and BYP_SEL and then observe the output from GPIO3, GPIO4, GPI5 and GPIO6 on logic analyzer. In Figure 6 it can be observed that when there is a change in the sel and byp signals, then there is a corresponding change in the frequency of the switching of the counter depicting that the PLL is being driven by 2 different clock sources.

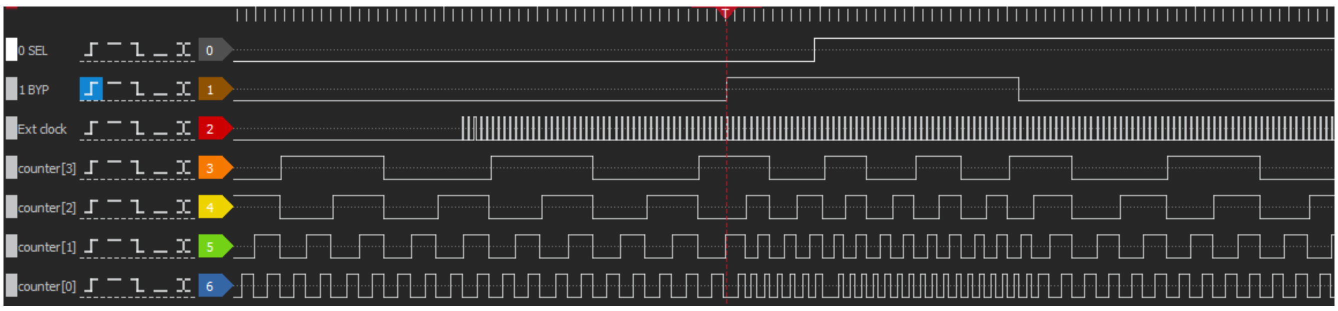


Figure 6: Change in sel and byp line to produce PLL_CLK of different frequencies

8. Conclusion

This application note shows how to drive the PLL using an External Clock (GPIO2) via REF_CLK_EXT or using the internal Oscillator on board via REF_CLK_OSC. This application note also focuses on how to set the values of different PLL parameters to achieve the desirable frequency when BYP is set to 0. This procedure can be utilized for any design. This testcase is available for download (AN-FG-006 Drive PLL from Oscillator.ffpga). If interested, please contact the ForgeFPGA Business Support Team.

9. Revision History

Revision	Date	Description
1.0	Mar 3, 2022	Initial release.
2.0	Feb 22, 2024	Updated changes according to BB revision
3.0	Jul 17,2024	Updated as per ForgeFPGA Workshop v6.43

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