How to use different Clocking Networks

SLG47921V

Abstract

This application shows how to drive the use the various clocks – on chip Oscillator, PLL and External Clock using SLG47921V ForgeFPGA and Deluxe Development Board. This document is supported by design files listed in the References Section.

Contents

© 2026 Renesas Electronics

1. Introduction

The FPGA Core has four clock domains in total. Each of the two tiles has two clock domains, Clock_0 and Clock_1. Each of them can use a unique clock or use the same clock, resulting in a total of up to four unique clocks being available. Tiles can exchange clocks or use each clock independently (depending on user application).

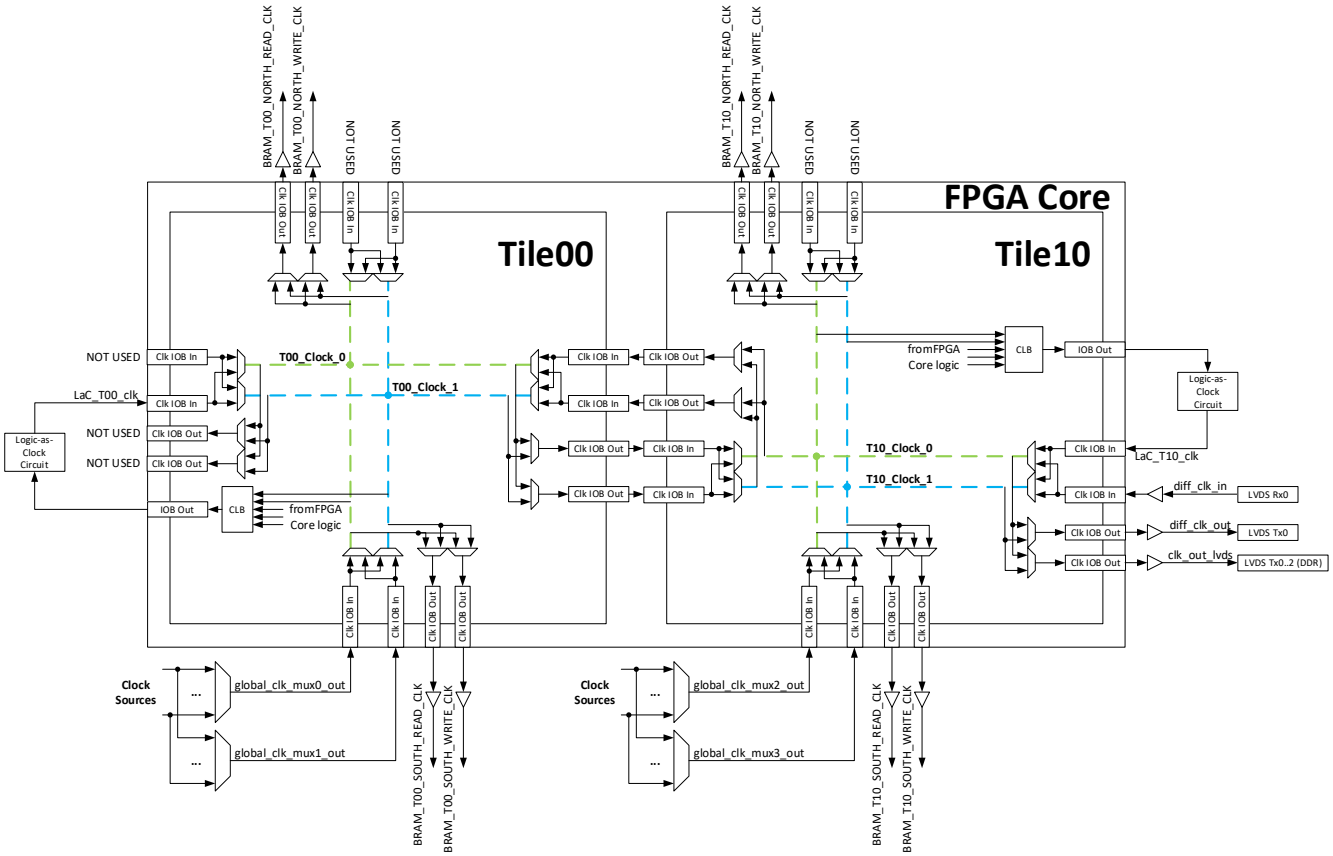


Figure 1.Clock Network

Clocks to the FPGA Core are provided via dedicated clock IOBs which are located as 2 IN + 2 OUT on each side of each tile. Clock IOBs along the sides where tiles are interconnected are used for inter-tile clock connections (defined by user logic). Therefore, 6 INs and 6 OUTs are available for use for external connections.

The core is divided into four directions:

North side

The BRAM North (Tile00 and Tile10 respectively) read and write clocks are connected to clock outputs.

Two clock inputs on the North side of each tile are occupied by the input clocks from the global clock distribution network (see Figure 2).

Tile00 West side

Logic-as-Clock (Tile00) is connected to clock input. For more details see 1.3.

Tile10 East side

Logic-as-Clock (Tile10) is connected to clock input. For more details see 1.3.

LVDS received differential clock is connected to dedicated clock input.

Two clock outputs are occupied by two clocks which are required by the LVDS transmit side:

- Transmit differential clocks
- Clock for DDR circuitry of LVDS

South side

The BRAM South (Tile00 and Tile10 respectively) read and write clocks are connected to clock outputs.

Global clock distribution network

Each tile has two separate clock IOB inputs fed by clock MUXes to independently select the clock source for each clock input. Clocks for the FPGA Core are provided symmetrically to all used tiles to obtain identical timing characteristics. A balanced clock tree is used.

Clock source selection is statically defined by Global Clock MUX Select option in SW individually for each clock MUX.

Figure 2 shows the global clock distribution network:

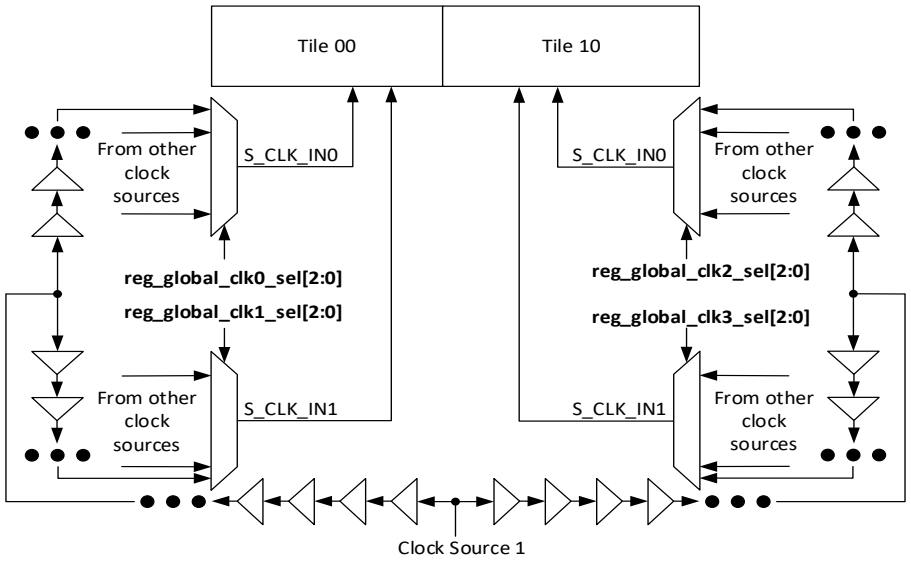


Figure 2.Global Clock Distribution Network

Sources available in the Global Clock Distribution Network consist of:

- PLL0 POSTDIV0 out clock
- PLL0 POSTDIV1 out clock
- PLL1 Out clock
- OSC POSTDIV0 out clock
- OSC POSTDIV1 out clock

1.1 On-Chip Oscillator

The SLG47920/SLG47921 has a High-frequency on-board oscillator for use within the high-density digital FPGA Core. During operating in either of the low-power modes of the FPGA Core, the high-frequency OSC is disabled. When the OSC_CTRL_EN signal is pulled high, there is a delay in receiving the signal at OSC_CLK. This delay allows the signal to stabilize, getting rid of any glitches in the output from the start. It has the following operating features:

- Two operating modes: high frequency (50 MHz) and low frequency (3 MHz)
- The ability to power-down the oscillator with the OSC_EN signal
- Ready signal to indicate that the output clock is stable

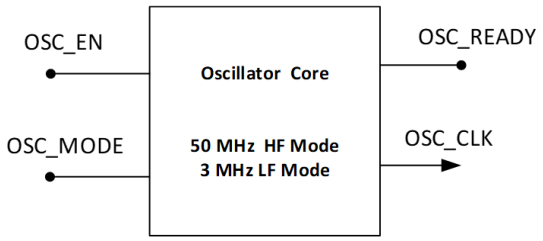


Figure 3.High-Frequency Oscillator Block Diagram

Signal Descriptions

Control inputs are derived from the FPGA Core.

- OSC_EN – active HIGH enable signal for the oscillator
- OSC_MODE – High/Low Frequency Mode selection. A HIGH-level input corresponds to 50 MHz and a LOW level corresponds to 3 MHz
- OSC_CLK – buffered oscillator clock output. It can be a clock source for PLLs or Oscillator Post Divider. It is connected to Global Clock Distribution Network only through Post Divider
- OSC_READY – outputs a HIGH level on this signal to indicate that the oscillator frequency is stable

Oscillator Post Divider

This circuit is intended to provide a low-frequency clock by dividing the Oscillator clock. The following functionality is implemented:

- Divide the Oscillator clock by factor of N in the range 1-128 (step with power of 2)
- Two divided output clocks OUT0/1 (each selected independently).
- IOB to select the division factor for each clock output individually.
- The ability to output the raw Oscillator clock frequency.
- Glitch-free and fast switching between different division factors.
- Two flags to indicate that the output clock is settled for each POSTDIV_OUT0/1

The Oscillator's post divider implementation is shown in Figure 4.

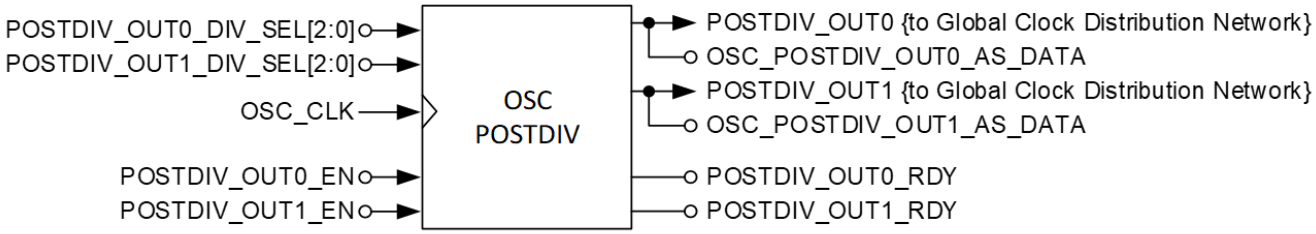


Figure 4.Oscillator Post Divider

Oscillator Post Divider outputs are connected to the Global Clock Distribution Network and at the same time – connected to dedicated IOBs of the FPGA Fabric to be used as low-frequency control signals if needed.

Control inputs (one each for OUT0 and OUT1) are derived from the FPGA Core.

- POSTDIV_OUT0/1_EN – Active HIGH enable signals for the Oscillator Post Divider Out0/1
- POSTDIV_OUT0/1_DIV_SEL – signals to select division ratio for Out0/1 of Oscillator Post Divider (
- POSTDIV_OUT0/1 – Oscillator Post Divider Out0/1 clock outputs. They are connected to the Global Clock Distribution Network.
- OSC_POSTDIV_OUT0/1_AS_DATA – Oscillator Post Divider Out0/1 clocks which are fed into the FPGA Core as data signals
- POSTDIV_OUT0/1_READY – a HIGH level is output on these signals to indicate that the Oscillator Post Divider Out0/1 is stable

Oscillator Post Divider OUT0's output frequency might be connected to a dedicated GPIO10 using the selection ***OUT0 fout to GPIO10*** to be able to output the clock for any external devices.

POSTDIV_OUT0/1_EN can be applied synchronously or asynchronously (relative to the POSTDIV_OUT0/1 clock) depending on ***OUT0/1 Enable Mode*** selection.

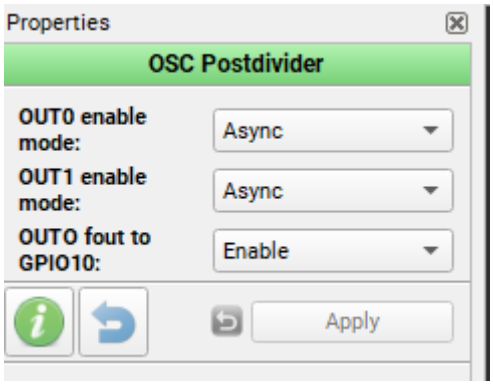


Figure 5.OSC Post Divider Settings in the software

Figure 5 displays the settings for OSC Post Divider in the GoConfigure Software that needs to be set to observe the output as needed.

1.2 Phase-Locked Loop (PLL)

The SLG47920/SLG47921 includes two low-power, wide input, and output integer phase-locked loops (PLLs) for use in applications requiring various frequencies. To maintain a low-power state the PLL has a power-down option for power management purposes. The input reference clock for the PLLs can either come from the internal oscillator, or from an external clock routed through GPIO6 (for PLL0) or GPIO7 (for PLL1), or the LVDS differential clock. Output clocks from both PLLs are available for FPGA Core use through the Global Clock Distribution Network.

PLL0 uses two external POSTDIV circuits instead of the internal POSTDIV circuit and produces two in-phase output clocks called PLL0_FOUT0 and PLL0_FOUT1. The external POSTDIV circuit is implemented in such a way that it is possible to predict the output phase shift between PLL0 FREF and PLL0_FOUT0/1. On the contrary, PLL1 uses the internal POSTDIV circuit and provides a single output frequency called PLL1_FOUT, in which the phase shift relative to PLL1 FREF is not predictable.

PLL0 is fully controlled from IOBs on the FPGA Core and PLL1 is controlled with a combination of static configuration (set with SW) and control signals from IOBs. PLL0_FOUT0 and PLL1_FOUT's output clocks are connected to dedicated GPIO8/9 with dedicated SW options enabled (***PLL FOUT0 to GPIO8*** / ***PLL FOUT to GPIO9***) to output the clock for any external devices.

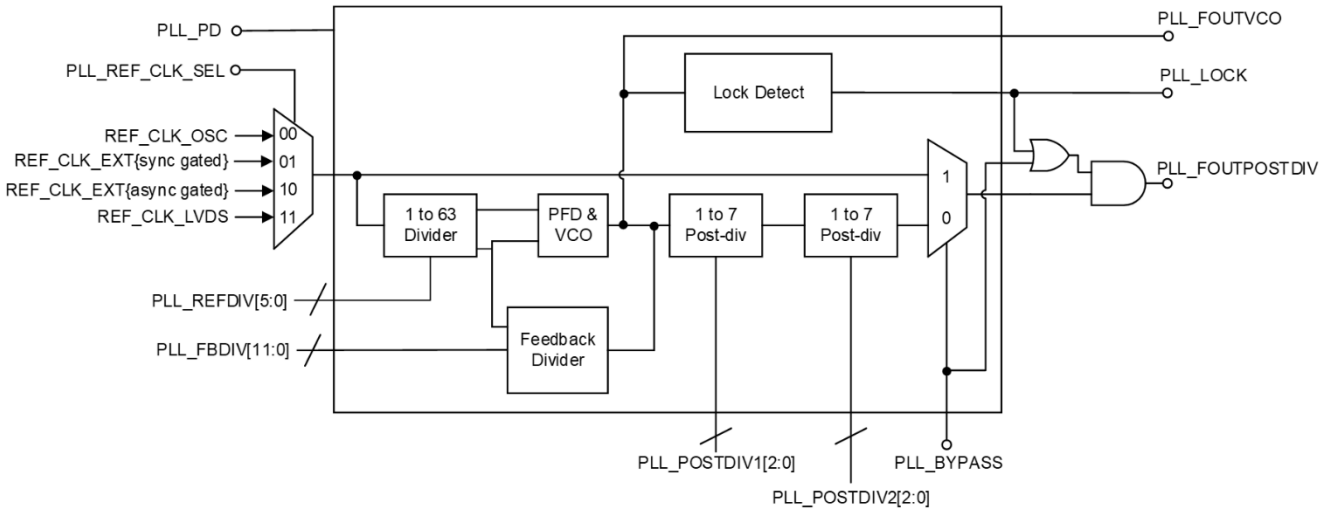


Figure 6.Phase-Locked Loop Block Diagram

The behavior of the SLG47920/SLG47921'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 or on-chip OSC, chosen through PLL_REF_CLK_SEL:

$$PLL\_FOUTVCO = (f_{reference\_clock} \times PLL\_FBDIV) / PLL\_REFDIV$$

$$PLL_{FOUTPOSTDIV} = \frac{f_{reference_{clock}} \times PLL_{FBDIV}}{PLL\_REFDIV \times PLL\_POSTDIV1 \times PLL_{POSTDIV2}}$$

Signal Descriptions

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

- PLL0/1_REF_CLK_SEL – selects the PLL Input Clock source between the internal OSC, external clock from GPIO6 or LVDS Rx Differential Clock and GPIO7 for PLL1
- PLL0/1_BYPASS – active HIGH signal that asserts a direct path between the FREF and PLL0/1_FOUT0
- PLL0/1_REFDIV [5:0] – sets the reference divide value (range 1 to 63)
- PLL0/1_FBDIV [11:0] – sets the PLL Feedback Divider value (range 16 to 400)
- PLL0/1_POSTDIV1_OUT0[2:0] & PLL0/1_POSTDIV2_OUT0 [2:1] – the two stages of Post Dividers are used to divide down the VCO Frequency before the FOUT0 clock output. Each stage has options for division from 1 to 7. The total division ratio is POSTDIV1 * POSTDIV2. This set of parameters is shared between PLL's internal post divider and external PLL Postdiv Out0
- PLL0_POSTDIV1_OUT1[2:0] & PLL0_POSTDIV2_OUT1 [2:1] – the two stages of external PLL Postdiv Out1 are used to divide down the VCO Frequency before the FOUT0 clock output. Each stage has options for division from 1 to 7. Total division ratio is POSTDIV1 * POSTDIV2
- PLL0_FOUT0/PLL0_FOUT1 – output clocks of PLL0
- PLL1_FOUT – output clocks of PLL1
- PLL0/1_LOCK – lock signal
- PLL0/1_RDY – ready signals, goes HIGH same time as PLL0/1_FOUT0/1 becomes available for user

Power inputs determine the existing power state of the PLL. Enable inputs are used to enable the different clock outputs that can be used in the FPGA Core and can lower power consumption when properly utilized.

All power and Enable inputs are connected to the FPGA Core.

- PLL0/1_EN – power enable for PLL0 and PLL1. Active HIGH

To learn more about PLL and its connections, see the datasheet.

The user can use the in-built PLL Configurator to calculate the output frequency for PLL0_fout0/1 and PLL1_fout. It helps you calculate the output frequency for different parameter inputs and updates the I/O Planner in the software. The configurator also generates the Verilog code for the associated PLL with the respective input parameters (see Figure 7). See the *Software User Guide for more information* [4]

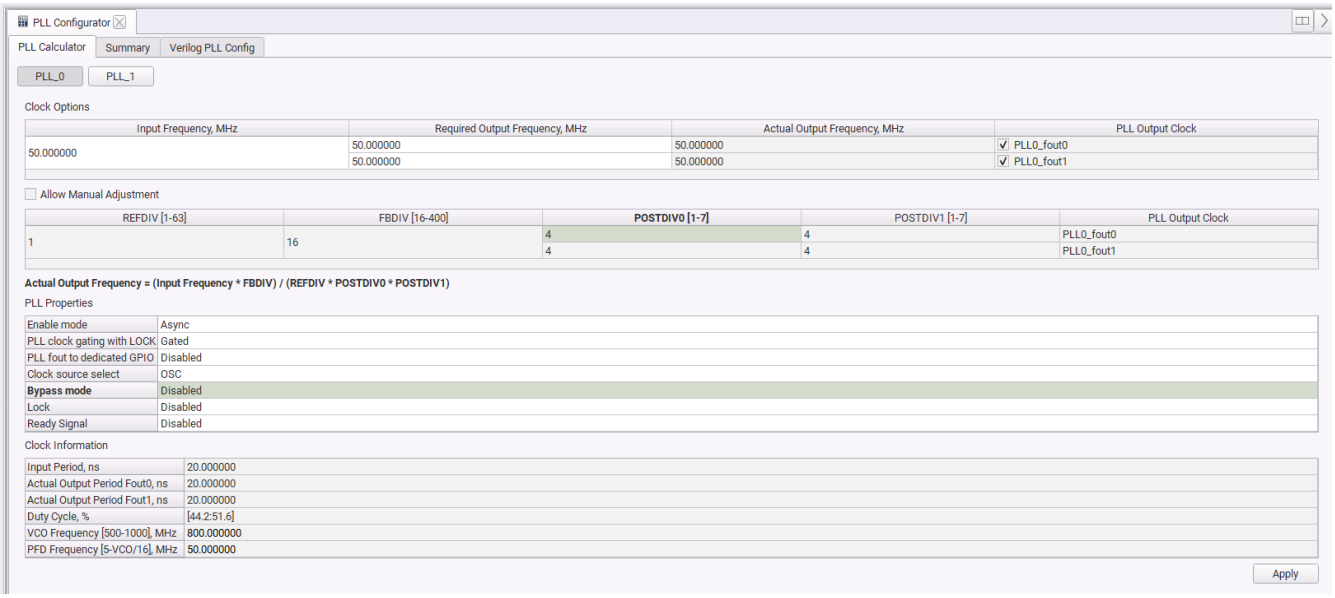


Figure 7.PLL Configurator

1.3 Logic - As – Clock (LaC)

In the SLG47920/SLG47921 FPGA Core, user logic signals can be utilized as a clock signal through a logic-as-clock path. For this purpose, the user signal which is used as the clock should output through Logic-IOB (LAC_T00/10_OUT) and then loop back into the FPGA core tile as the clock (LAC_T00/10_IN) through clock-IOB (see Figure 8). This is known as Logic-as-Clock circuitry (LaC Circuitry). There are two LaC paths available (see Figure 1).

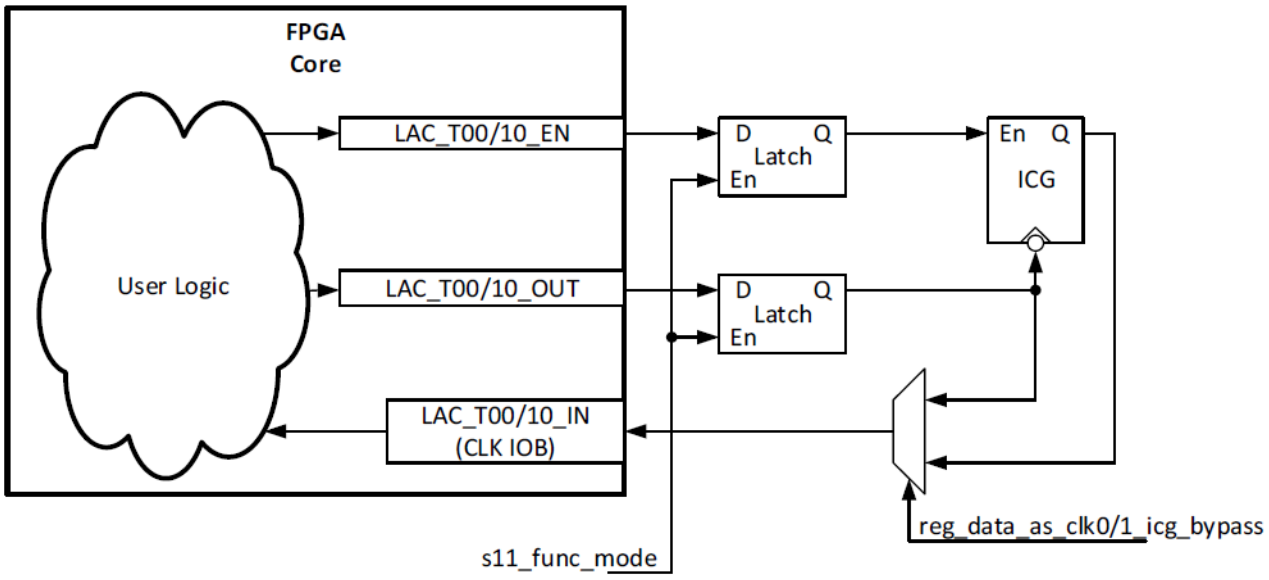


Figure 8.Logic-As-Clock Usage

To activate Logic-as-clock paths there are dedicated IOBs: LAC_T00/10_EN.

With Logic-as-Clock circuitry there are two ways to transform the data signal into a clock signal:

- The clock signal is generated inside the FPGA Core tile (for example OSC/8) and then fed through LaC path
- The clock signal is fed through any GPIO to the FPGA Core tile, transfers it without processing (in an asynchronous manner) and is fed through LaC0 path.

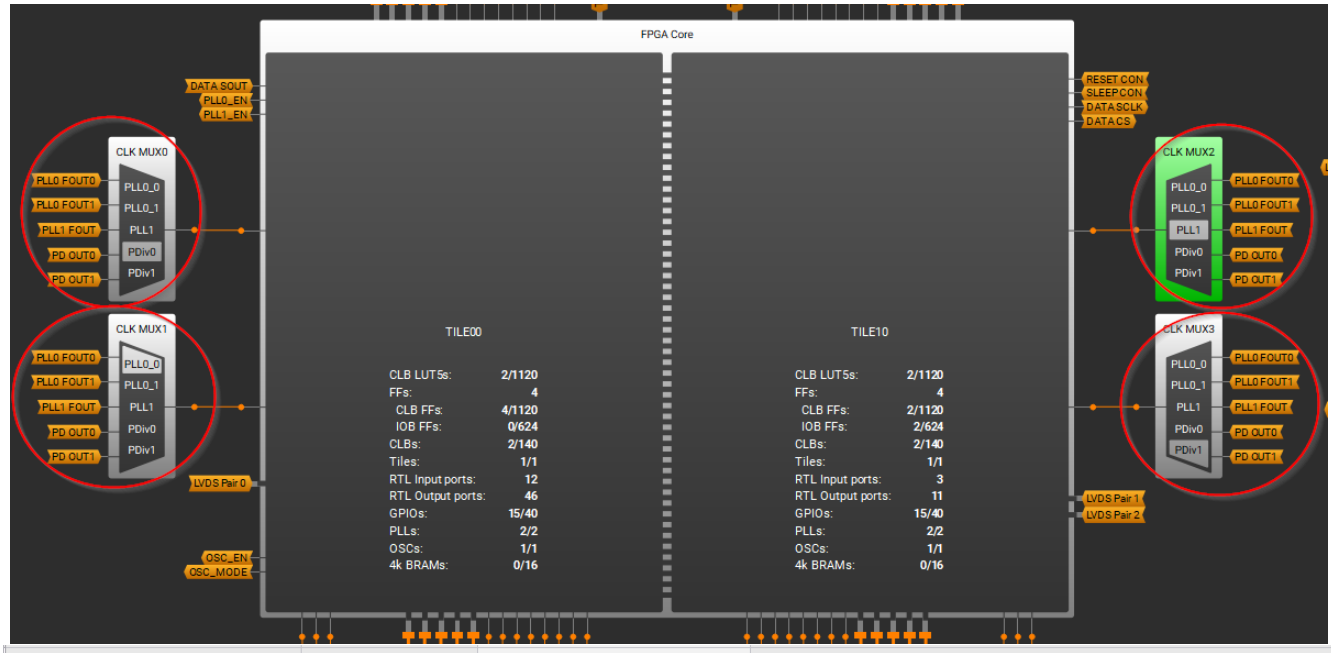
The LaC circuit provides the following functionality:

- User logic signals as clocks
- Turn LaC clock on/off using ICG (integrated clock gating cell)
- Optionally use LaC without ICG (defined by Clock Path option in SW)
- Settle LaC clock in its last state (HIGH or LOW) when exiting Functional Mode

FUNCTION	DIRECTION	PORT	DESCRIPTION
LAC_T00_EN	Output	o_lac0_ctrl_en	Tile00 Logic-as-Clock enable signal (if ICG is used)
LAC_T00_IN	Input	o_lac0_clk	Tile00 Logic-as-Clock Input
LAC_T10_OUT	Output	o_lac0_clk	Tile10 Logic-as-Clock output signal (to be promoted to Clock)

Figure 9.LaC Connections in I/O Planner

Figure 10 shows the clock connections between the clock muxes to the 2 tiles, specifically how each clock mux is connected to different clock source. It must be specified in the ForgeFPGA Workshop as well as the I/O Planner.



global_clk_mux_out[0]	Input	i_osc_clk	Tile00 Clock Input 0 from Global Clock Network
global_clk_mux_out[1]	Input	i_pll0_fout0	Tile00 Clock Input 1 from Global Clock Network
global_clk_mux_out[2]	Input	i_pll1_clk	Tile10 Clock Input 0 from Global Clock Network
global_clk_mux_out[3]	Input		Tile10 Clock Input 1 from Global Clock Network

Figure 10.Clock Mux connections

2. Requirements

- SLG47920/21V Device
- FPGA Deluxe Development Board with USB cable and power supply
- ForgeFPGA Socket Adaptor Board
- Latest Revision of ForgeFPGA Workshop software

3. Verilog Code

Shown below is the Clock configuration mentioned in the module named Clock. The Verilog code for How to Use Clock can be found in the complete design example available for download in the References section.

```
// OSC CONFIGURATIONS
  assign o_osc_en                = 1;
  assign o_postdiv_out0_en       = 1;
  assign o_osc_mode              = i_osc_mode;                    //from gpio  1 = 50 MHz ; 0 = 3 MHz
  assign o_osc_rdy               = i_osc_rdy;                     // to check value on gpio
  assign o_postdiv_out0_rdy      = i_postdiv_out0_rdy;          // to check value on gpio
  assign o_postdiv_out0_div_sel = 3'b000;

// PLL CONFIGURATIONS
  assign o_pll0_rdy            = i_pll0_rdy;                 // to check status on gpio
  assign o_pll0_lock           = i_pll0_lock;                // to check status on gpio
  assign o_pll0_ctrl_en        = 1;
  assign o_pll0_ref_clk_sel  = i_pll0_ref_clk_sel;           // ref clk --> gpio 6
  assign o_pll0_bypass         = i_pll0_bypass;
  assign pll0_refdiv             = 6'b00_0001;         //1
  assign pll0_fbdiv              = 12'b0000_0001_0010;  //18
  assign pll0_postdiv0_out0  = 3'b011;             //3
  assign pll0_postdiv1_out0  = 3'b011;             //3

  assign o_pll1_rdy          = i_pll1_rdy;                 // to check status on gpio
  assign o_pll1_lock         = i_pll1_lock;                // to check status on gpio
  assign o_pll1_ctrl_en      = 1;

// LaC CONFIGURATIONS

  assign o_lac0_ctrl_en      = 1 ;
  assign o_lac0_clk          = i_lac0_clk_source;           // to re-route the input clock
```

4. Floorplan

Figure 11. 2-tile Floorplan

The Floorplan tab in the FPGA Editor shows the placement of the CLBs, FFs and their connections to IOB blocks between the two tiles of SLG47910 forming together one SLG47921V.Hovering over the connection tabs will highlight where the connection leads to for further understanding.
User can zoom in using the footer controls or by pressing CTRL + Scroll on the mousewheel.

5. Design Steps

1. Launch the latest version of the Go Configure Software Hub. Select the SLG47921V device and the ForgeFPGA Workshop software will load.
2. Download the design example AN-FG-021 How to use different Clock Networks.ffpga.
3. Open the AN-FG-021 How to use different Clock Networks.ffpga file after downloading.
4. Open the FPGA editor and review the Verilog code. The module name is Clock which covers the working of the OSC clock, PLL0/1 Clock and LaC0 clock.
5. Open the PLL Configurator and confirm the parameters for the desired clock output.
6. Open the IO planner tab on the FPGA editor and review the pin assignment. Ensure that the clock inputs to CLK MUX in the IO Planner (see Figure 10)
7. Select the Synthesize button on the lower left side of the FPGA Editor.
8. 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.
9. Click on the Floorplan tab and see the CLB utilization (see Figure 11). Press the Ctrl and the mouse wheel to zoom-in. Confirm that the IOs selected in the IO Planner are shown in the floorplan.
10. In the ForgeFPGA Workshop, locate the assigned CLK MUX Macrocell and open its properties. From the dropdown menu for each CLK MUX, select the associated clock input as specified in Figure 10.
11. Connect the Deluxe Development Board and attach it to Adaptor Board with the SLG47921V part in the socket on it. Click on the Debug button on the ForgeFPGA Workshop studio and select Emulation.
12. User can observe that each GPIO has been labeled as per their assigned functionality. Synchronous Generator has been used to generate external clock on GPIO6, GPIO7 for PLL_REF_EXT clocks. Similarly, GPIO17 is assigned a clock of 30.33MHz to observe the functionality of LaC.

13. Launch the Logic Analyzer from the top toolbar to view waveform outputs. Inside the Logic Analyzer, select the GPIOs of interest and run it on Auto Mode. At the same time, turn on the Synchronous Logic Generator to get the clock started on the GPIOs you want to observe.
14. User can observe PLL0_FOUT0_CLK on GPIO8, PLL1_FOUT_CLK on GPIO9 and OSC_CLK on GPIO10.

6. Results

Case 1. When the OSC is used as reference clock for PLL along with the set PLL parameters in the Verilog Code.

Osc_mode = 1 (50MHz)

o_pll0_ref_clk_sel	o_pll0_bypass	pll0_refdiv	pll0_fbdiv	pll0_postdiv0_out0	pll0_postdiv1_out0
00	0	1	18	3	3

o_pll1_ref_clk_sel	o_pll1_bypass	pll1_refdiv	pll1_fbdiv	pll1_postdiv0_out0	pll1_postdiv1_out0
00	0	1	16	5	6

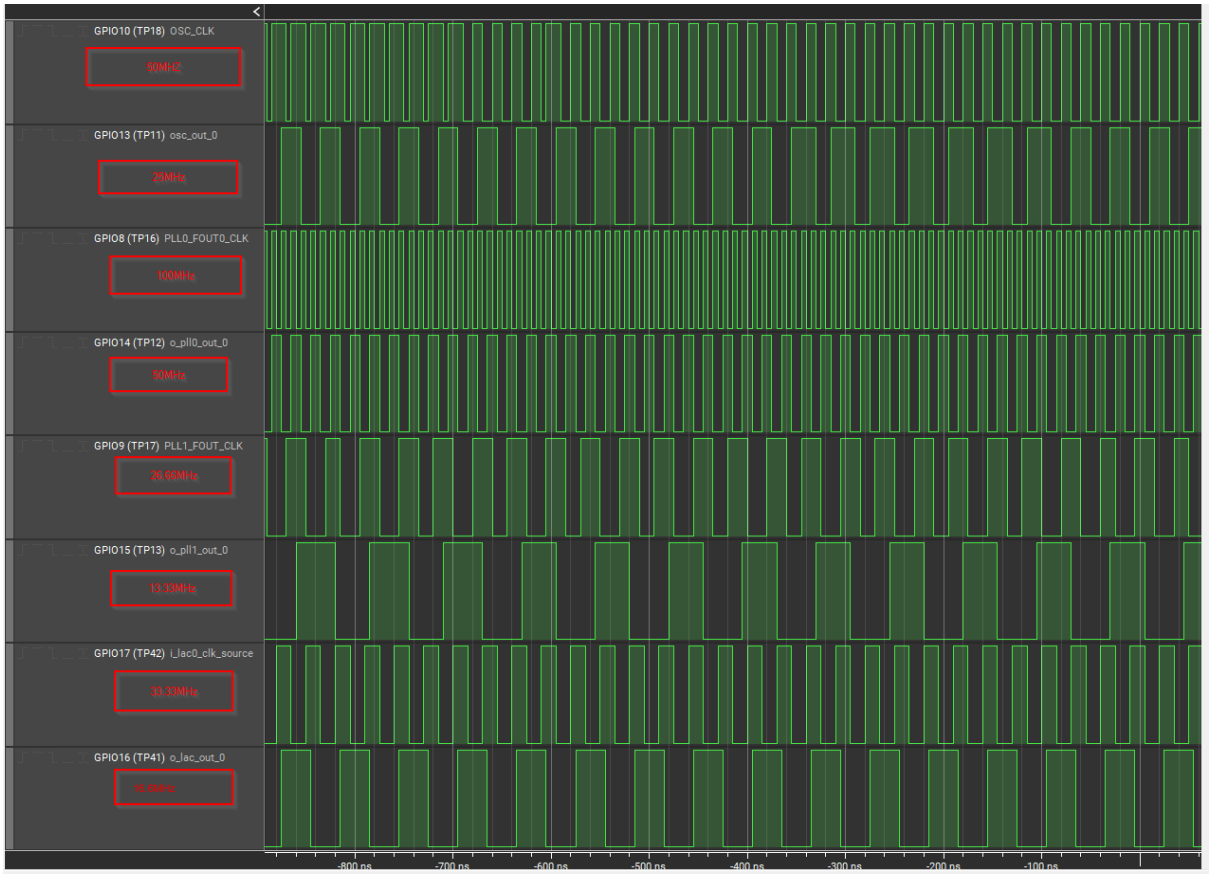


Figure 12. Logic Analyzer output for case 1

Case 2. Bypass mode is enabled for PLL0 and PLL1

o_pll0_ref_clk_sel	o_pll0_bypass	pll0_refdiv	pll0_fbdiv	pll0_postdiv0_out0	pll0_postdiv1_out0
00	**1**	1	18	3	3

o_pll1_ref_clk_sel	o_pll1_bypass	pll1_refdiv	pll1_fbdiv	pll1_postdiv0_out0	pll1_postdiv1_out0
00	**1**	1	16	5	6

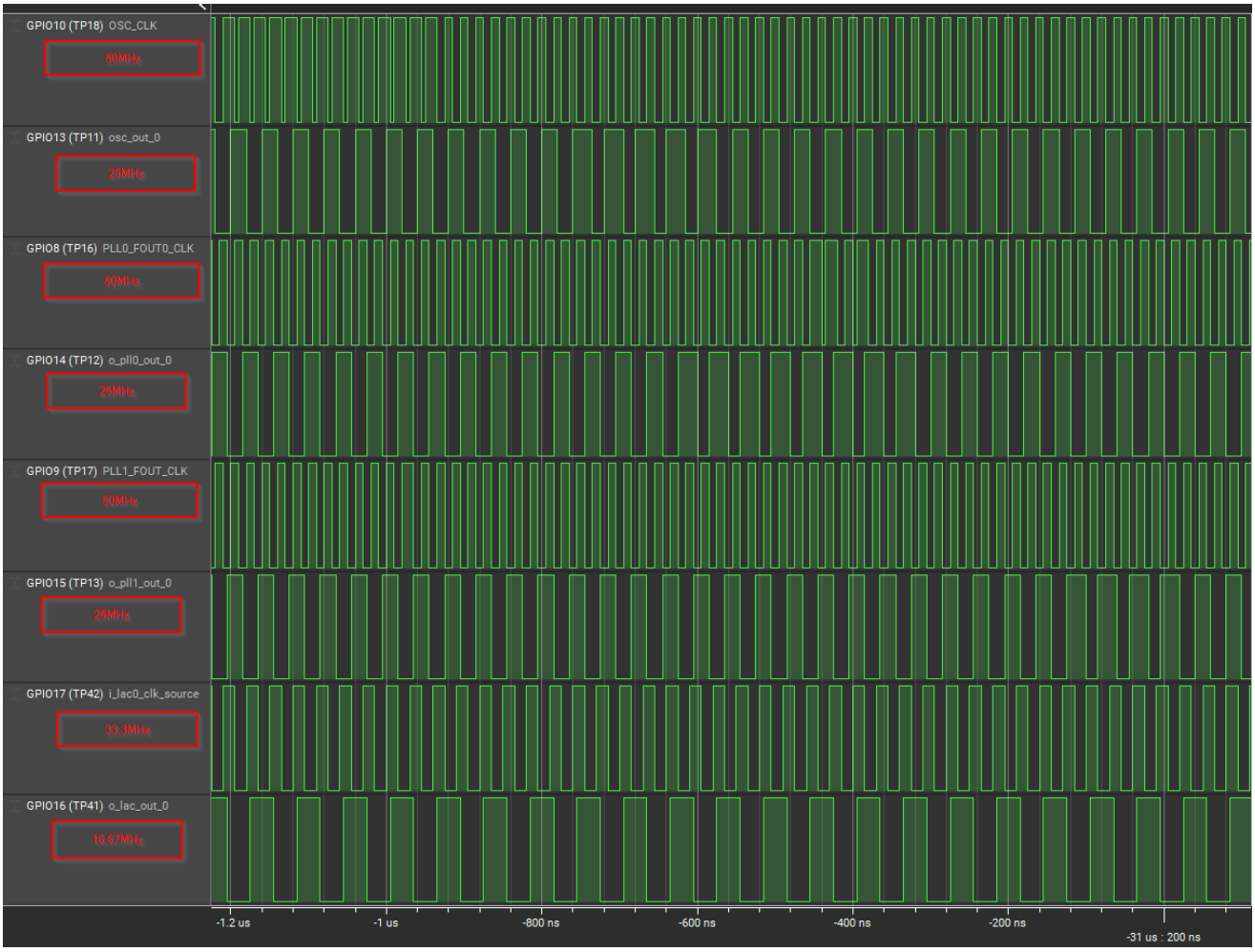


Figure 13. Logic Analyzer output for case 2

Case 3. When the OSC is used as reference clock for PLL along with the set PLL parameters in the Verilog Code.

Osc_mode = 0 (3 MHz)

o_pll0_ref_clk_sel	o_pll0_bypass	pll0_refdiv	pll0_fbdiv	pll0_postdiv0_out0	pll0_postdiv1_out0
00	0	1	18	3	3

o_pll1_ref_clk_sel	o_pll1_bypass	pll1_refdiv	pll1_fbdiv	pll1_postdiv0_out0	pll1_postdiv1_out0
00	0	1	16	5	6

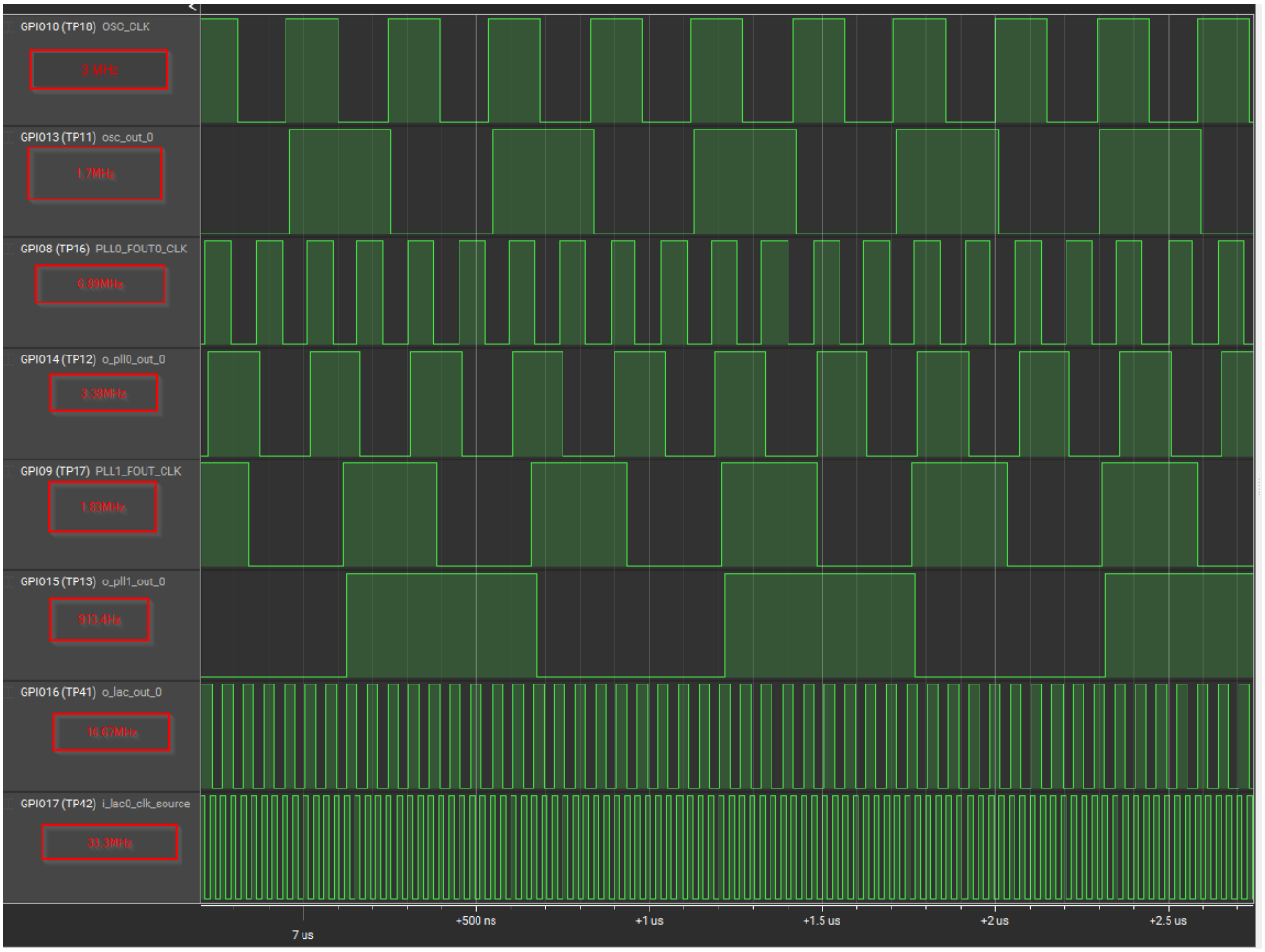


Figure 14. Logic Analyzer for Case 3

Case 4. PLL1 clock source is external through GPIO7 (100 MHz) for PLL1 for the same parameters. The Clock source must be changed from OSC to *Ext. Clk async. gated* in the properties and the Emulation re-run.

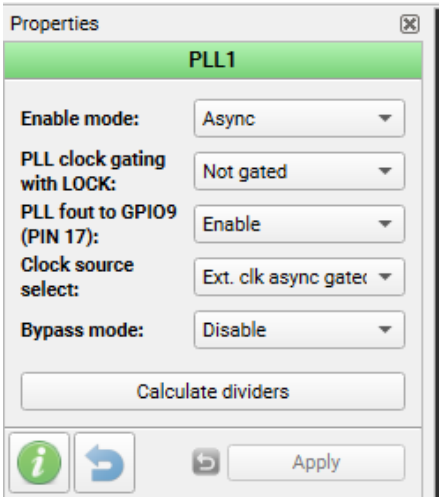


Figure 15. PLL1 Properties

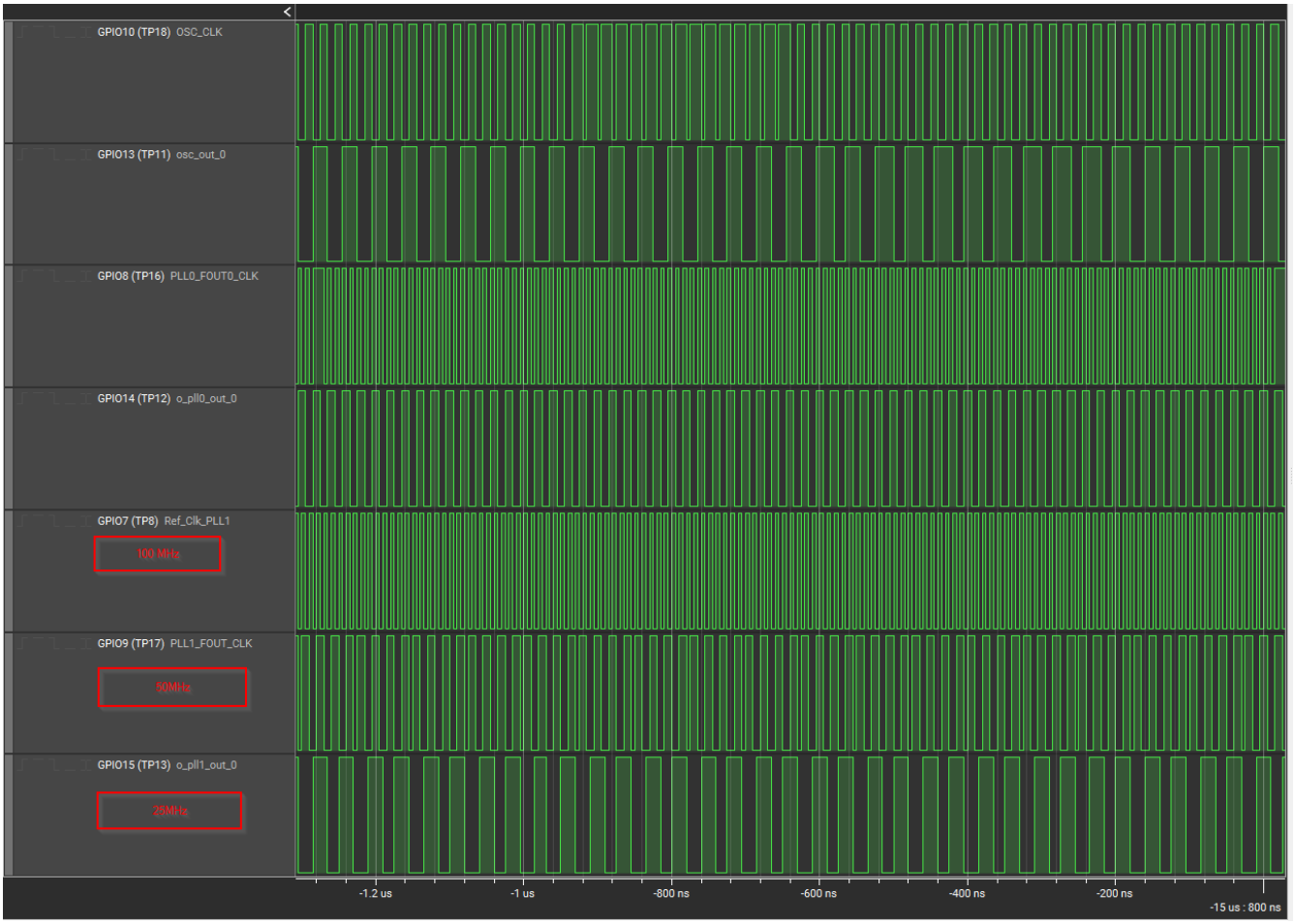


Figure 16. Logic Analyzer for Case 4

Case 5. PLL0 clock source is external through GPIO6 (33.3 MHz) for PLL0 for the same parameters. User needs to select 10 at GPIO18 and GPIO19 to reflect the change of ext. reference clock.

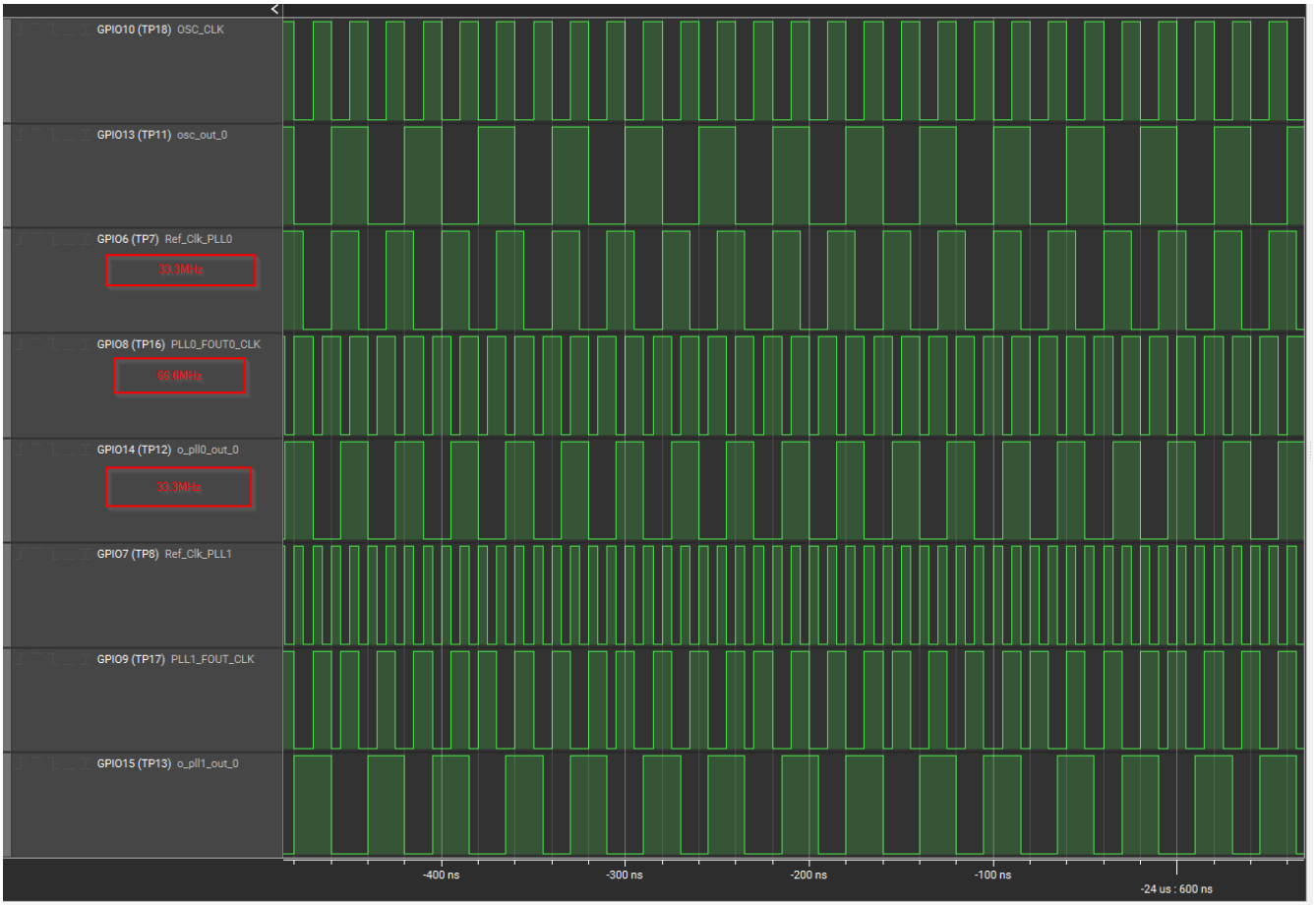


Figure 17. Logic Analyzer for Case 5

7. Conclusion

This Application Note focuses on how to use the few different clocks available on SLG47920/21 and how to activate it using the GoConfigure Software. The document explains the basic default features of PLL0/1, OSC and LaC00. This testcase is available for download (AN-FG-021 How to use different Clock Networks.ffpga). For more information, contact the ForgeFPGA Business Support Team.

8. 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
LaC	Logic-As-Clock

9. References

For related documents and software, please visit: ForgeFPGA. Download our free ForgeFPGA™ Designer software [1] to open the. ffpga files [2] and view the proposed circuit design.

[1] Go Configure Software Hub, Software Download and User Guide

[2] AN-FG-021 How to use different Clock Networks. ffpga, ForgeFPGA Design File

[3] ForgeFPGA SLG47920/21, Datasheet, Renesas Electronics

[4] ForgeFPGA Workshop User Guide, Renesas Electronics

10. Revision History

Revision	Date	Description
1.0	Jan 06,2026	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 Apr 2025)

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-data 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 are registered Trademarks are the property of their respective owners.

© 2025 Renesas Electronics Corporation. All rights reserved.