Configuring Clocks Demo Script

Introduction

This demonstration provides high-level instructions on configuring the clocks for the Zynq™ UltraScale+™ MPSoC.

Preparation

- Necessary files are located at `$TRAINING_PATH/MPSoC_Clocks/support` and `$TRAINING_PATH/CustEDIP`
  - `clk_demo_completer.tcl`
  - `completer_helper.tcl` (helper script containing common hardware operations)
- Required hardware: None
- Supporting materials:
  - *Zynq UltraScale+ MPSoC Technical Reference Guide* (UG1085)

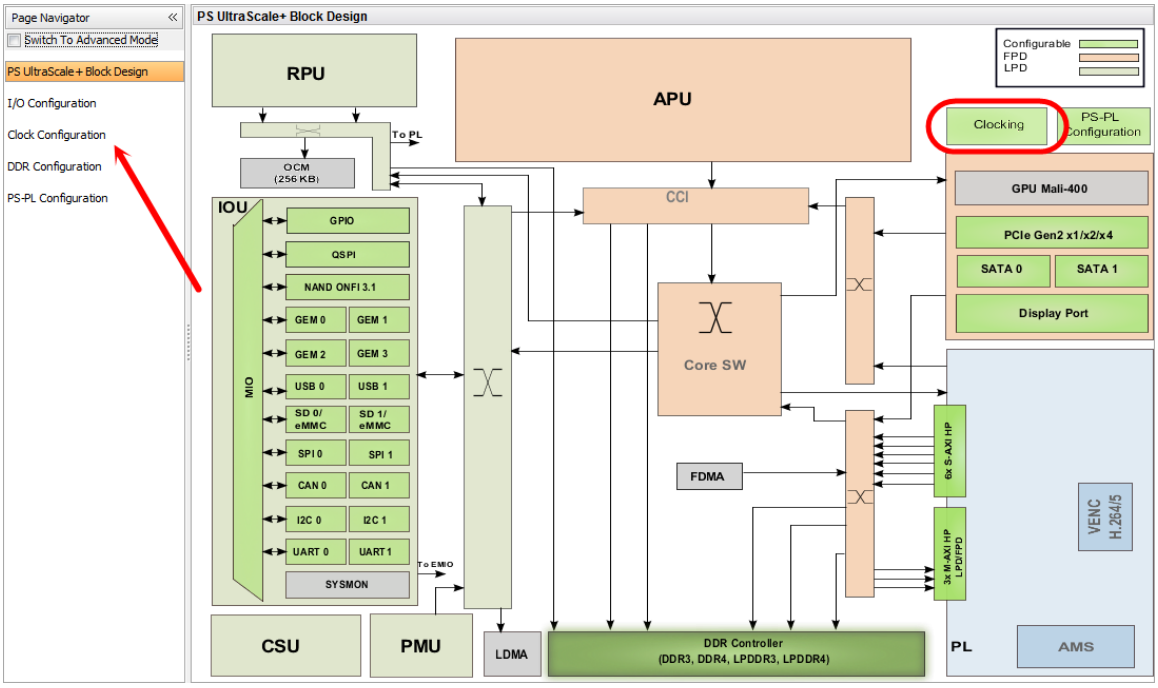
Configuring Clocks

Action with Description	Point of Emphasis and Key Takeaway
• Click the **Vivado** icon to launch the Vivado™ Design Suite.	

• From the Tcl command console:
`cd $::env(TRAINING_PATH)/MPSoC_Clocks/support`
• Load the Tcl script provided to build the project:
`source clk_demo_completer.tcl`
• Specify the default language, platform, and processor:
`use VHDL`
`use ZCU104`
`use APU`
• Build the hardware:
`make buildStartingPoint`
	• Builds a simple project with a Zynq UltraScale+ MPSoC device.

Action with Description	Point of Emphasis and Key Takeaway
• You will begin by looking at a number of the clocking options available in the PS.	

• Double-click the **Zynq UltraScale+ MPSoC** IP to open the Re-customize IP window.	• This will open the customization window where the following can be customized:
	  • I/O configuration
	  • Clock configuration
	  • DDR configuration
	  • PS-PL configuration
	• All green boxes are configurable.

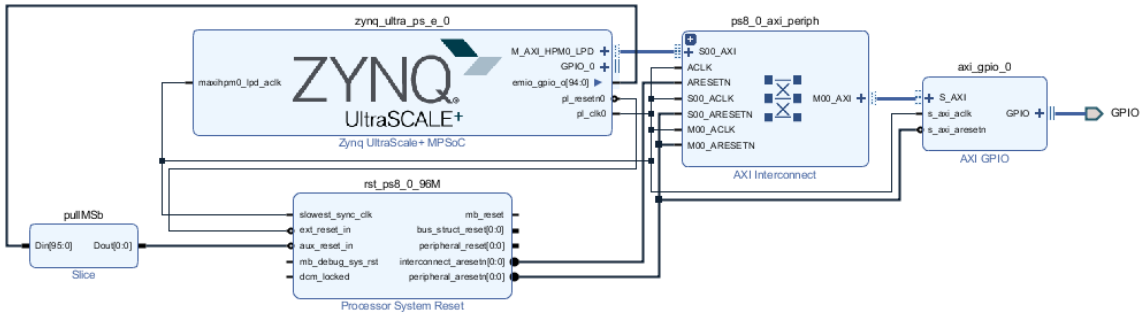
	

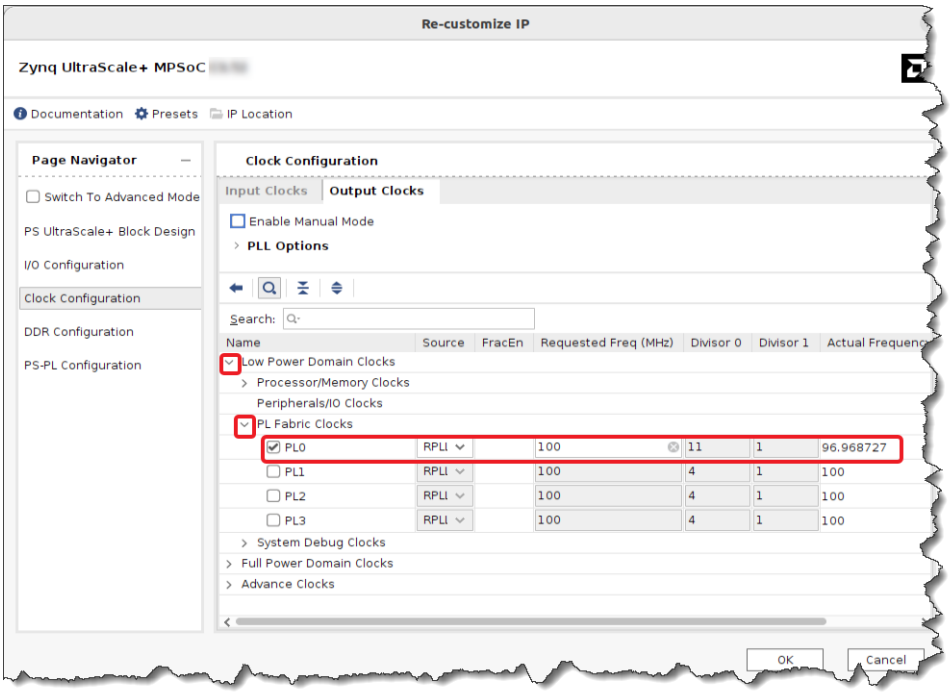
• Click **Clock Configuration** in the Page Navigator.	• There are three sections where the clocks can be configurable:
	  • PLL reference clocks
	  • Low-power domain clocks
	  • Full-power domain clocks

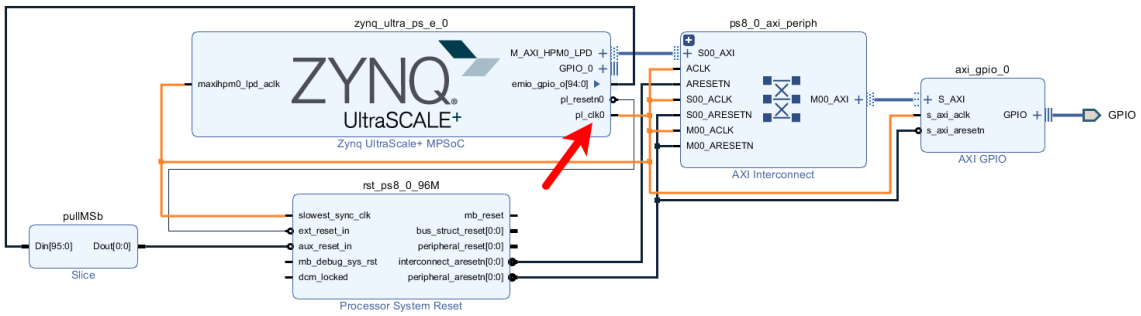
Action with Description	Point of Emphasis and Key Takeaway

• Point out that the Input Clocks are shown by default (notice the tabs across the top).
• Expand **Input reference frequency** to expose the PSS_REF_CLK.
• Switch to the Output Clocks tab by clicking it.
• Notice that the clocks are grouped into Low-Power, Full-Power, and Advance(d) clocks.
• Click the **Expand All** icon to expand all the fields.
• You may wish to maximize this view by dragging the corner of the window.
	• The PS supports eight clock inputs:
	  • PSS_REF - primary reference clock
	  • PSS_ALT_REF - optional clock via the MIO
	  • Video_CLK - optional clock via MIO
	  • AUX_CLK - five independent auxiliary clocks from the PL
	• Each input clock may be between 27 and 60 MHz.
	• The Zynq UltraScale+ MPSoC has five PLLs that generate the various clocks used in the PS subsystem:
	  • APU PLL (APLL) generates the clock for the APU.
	  • DDR PLL (DPLL) generates the clock for the DDR controller.
	  • I/O PLL (IOPLL) generates the clock for the peripheral I/Os.
	  • RPU PLL (RPLL) generates the clock for the RPU.
	  • Video PLL (VPLL) generates the clocks for the video blocks used in the PS subsystem.

• Review the contents of the **Low Power Domain Clocks.**	• I/O PLL (IOPLL) provides clocks for all low-speed peripherals and are part of the interconnect. • RPU PLL (RPLL) provides clocks for the Cortex™-R5 CPU and are part of the interconnect. **Note:** The PLL naming convention refers to RPU PLL as RPLL. However, RPU can be clocked using other PLLs as well. For example, the IOPLL can also clock the RPU. • **PL Fabric Clocks:** PL0, PL1, PL2, PL3.

Action with Description	Point of Emphasis and Key Takeaway
• Review the contents of the **Full Power Domain Clocks.**	• APU PLL (APLL) provides clocks for the Cortex-A53 CPU clock and part of the interconnect. • Video PLL (VPLL) provides clocks for the video I/O. • DDR PLL (DPLL) provides clocks for the DDR controller and part of the interconnect. • DDR PHY provides its own PHY PLL (PPPL) to provide clocks for the DDR PHY. **Note:** The PHY PLL is dedicated to the DDR interface and cannot be used as a clock source for other blocks in the PS.
• Review the contents of the Advance(d) Clocks.	• Less frequently used clocks are listed here and further subdivided into the Low-Power Domain and Full Power Domain
• Close the Re-Customize window without making changes (**Cancel**).	
Time to look at the rest of the provided design that was built when the demo was started. • This Tcl file will generated a block design with GPIO connects to the LED on the ZCU104 board. The GPIO is connected the clock and reset. • The block design should resemble the figure below:	
	

Action with Description	Point of Emphasis and Key Takeaway
• Double-click the **Zynq UltraScale+ MPSoC** IP. This will open up the customization window.	• This is done to check the default setting of the one clock that is emitted by the PS to the PL.
• Select the Clock Configuration from the Page Navigator. • Expand the **Lower Power Domain Clocks** > **PL Fabric Clocks**. • Notice that the PL0 provides a 100-MHz clock to the programmable logic by default. • Select the **PL0** entry in the Zynq UltraScale+ MPSoC block. • Click **OK** to exit.	• Explore the default settings for the PL clock connections.
	
• You should now be looking at the overall block diagram. • Click the net connected to **pl_clk0** on the PS.	• Explore how the tools will automatically connect the clock when only one clock is presented from the PS.

Action with Description	Point of Emphasis and Key Takeaway
• Question: What do you think happens when there is more than one clock available from the PS to the PL? • Answer: When the "Run Connection Automation" tool runs, it will try to connect everything to the lowest numbered clock. You need to explicitly indicate that you want to use another clock. • Question: Where would this clock need to be routed to if the new peripheral were using it? • Answer: To the peripheral, to the AXI interconnect, and possibly the reset block if it were the slowest of the available system clocks.	
	
• Close the Vivado Design Suite.	

Summary

In this demonstration, you have learned how to configure the primary input clocks and output clocks of a Zynq UltraScale+ MPSoC device. You have also learned how to reset the PL fabric using the EMIO.