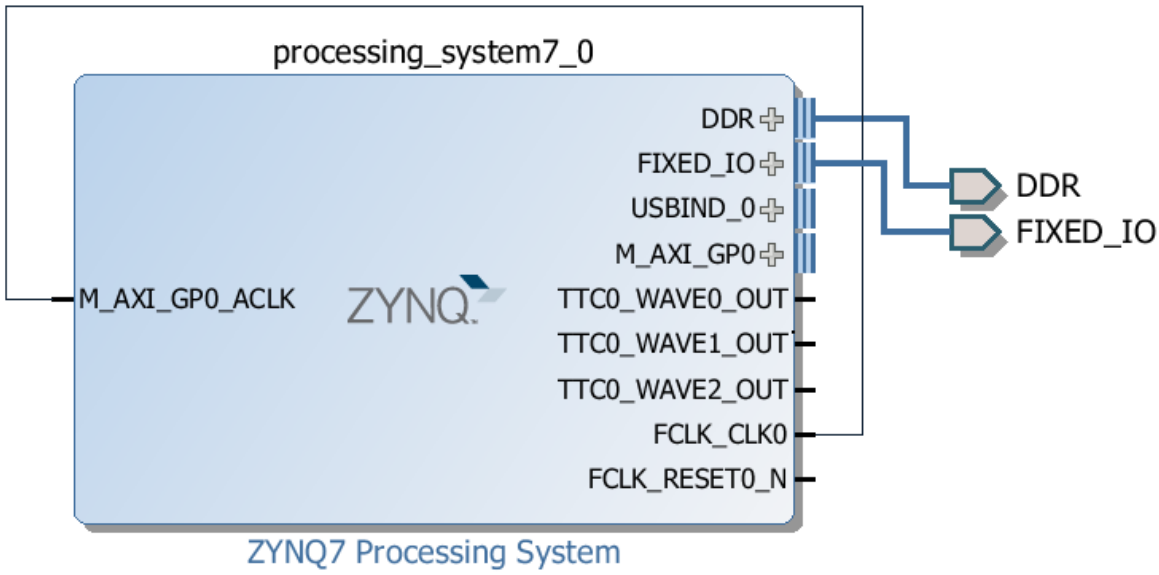
FPGAdeveloper

Creating a Base System for the Zynq in Vivado

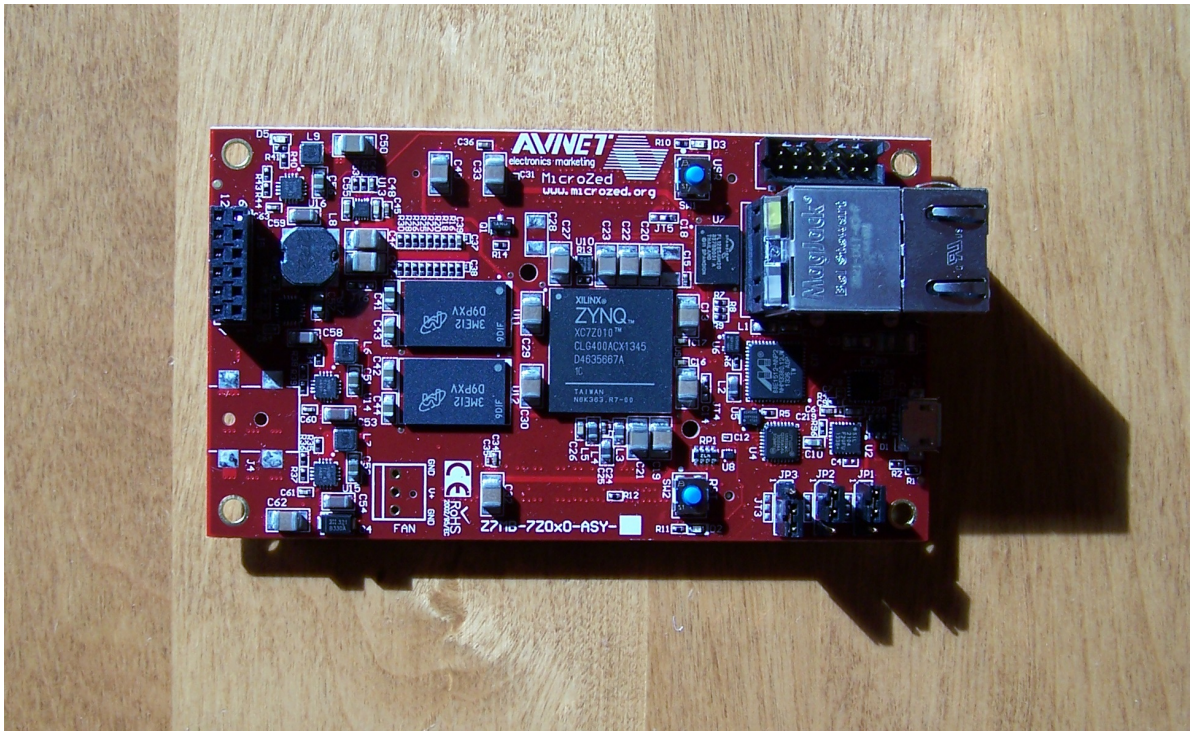
by Jeff Johnson | Jul 31, 2014 | Vivado | 8 comments



5 Votes

Tutorial Overview

In the ISE/EDK tools, we'd use the Base System Builder to generate a base project for a particular hardware platform. Now with Vivado, the process is a little different but we have more control in how things are setup and we still benefit from some powerful automation features. In this tutorial we'll create a base design for the Zynq in Vivado and we'll use the MicroZed board as the hardware platform.



Requirements

Before following this tutorial, you'll need the following:

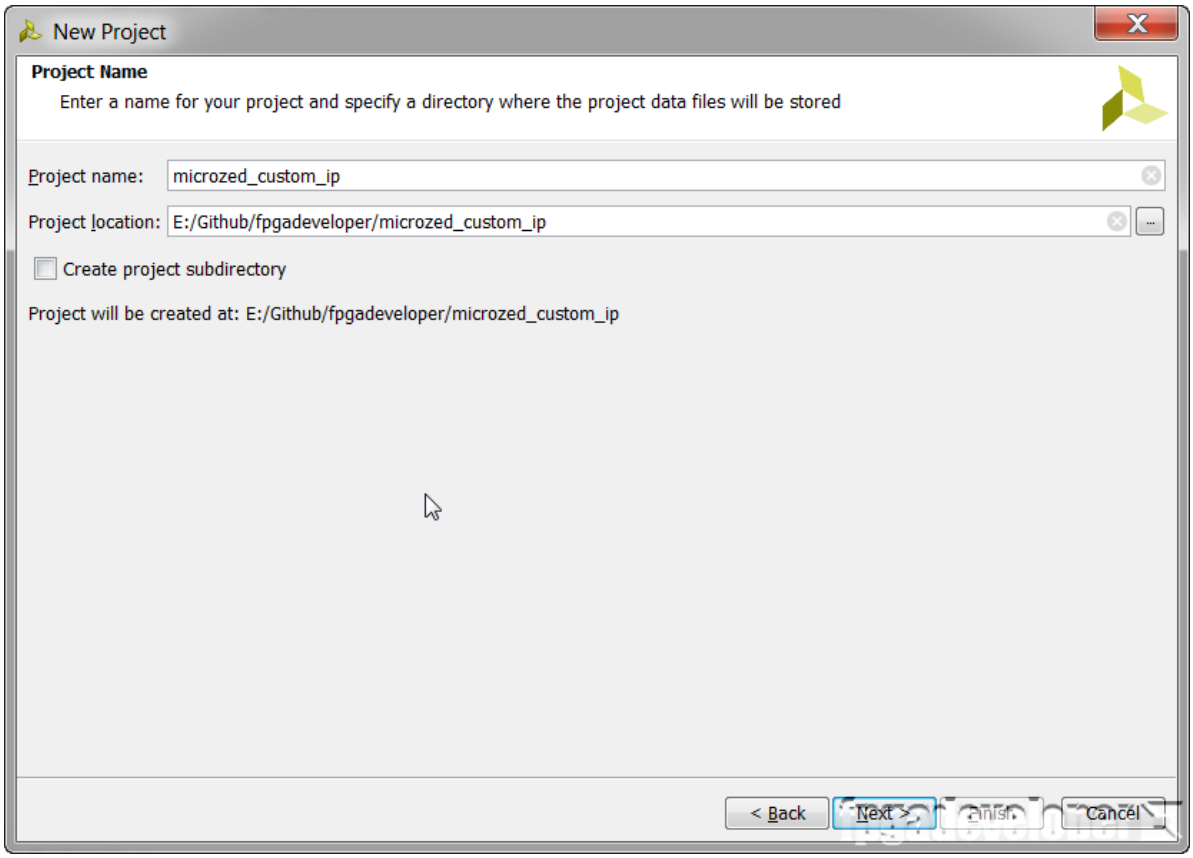
- Vivado 2014.2
- MicroZed
- Platform Cable USB II (or equivalent JTAG programmer)

Create a new Vivado project

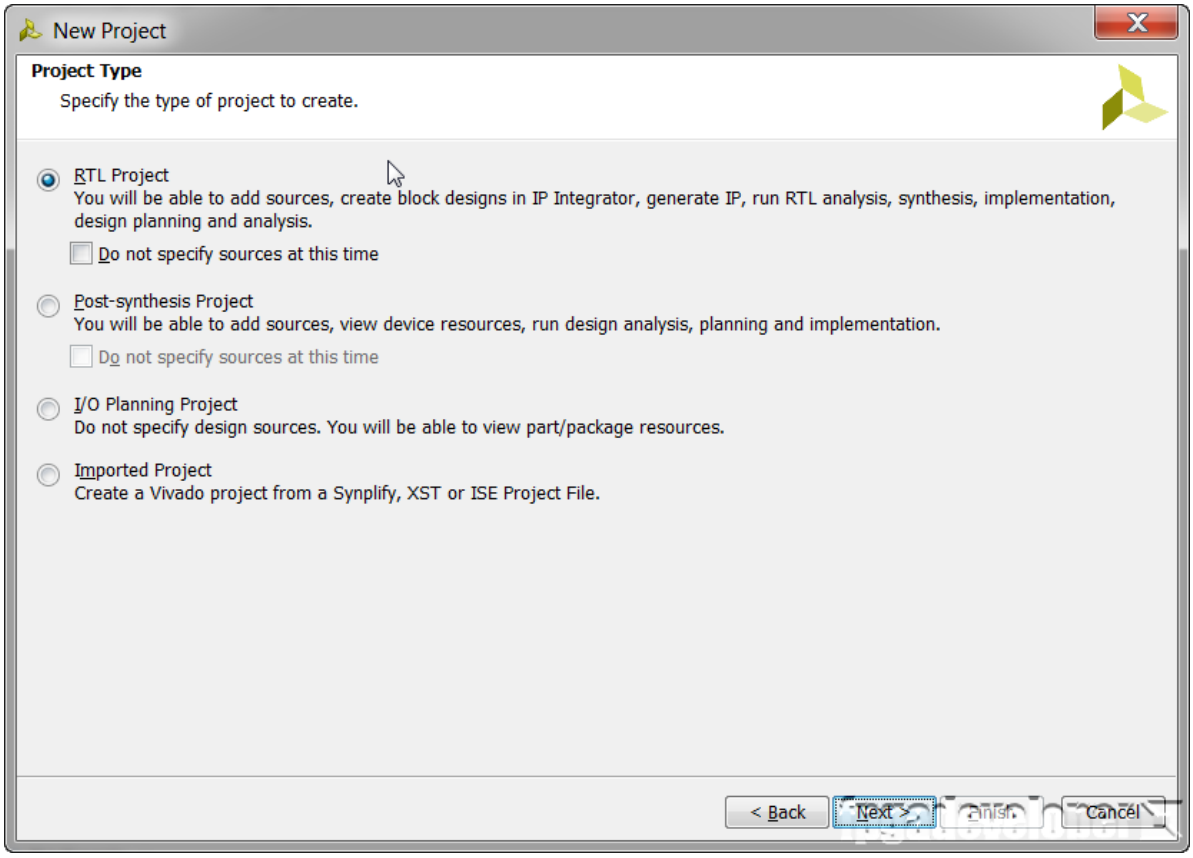
Follow these steps to create a new project in Vivado:

1. Open Vivado. From the welcome screen, click "Create New Project".

2. Specify a folder for the project. I've created a folder named "microzed_custom_ip". Click "Next".



3. For the Project Type window, choose “RTL Project”. Click “Next”.

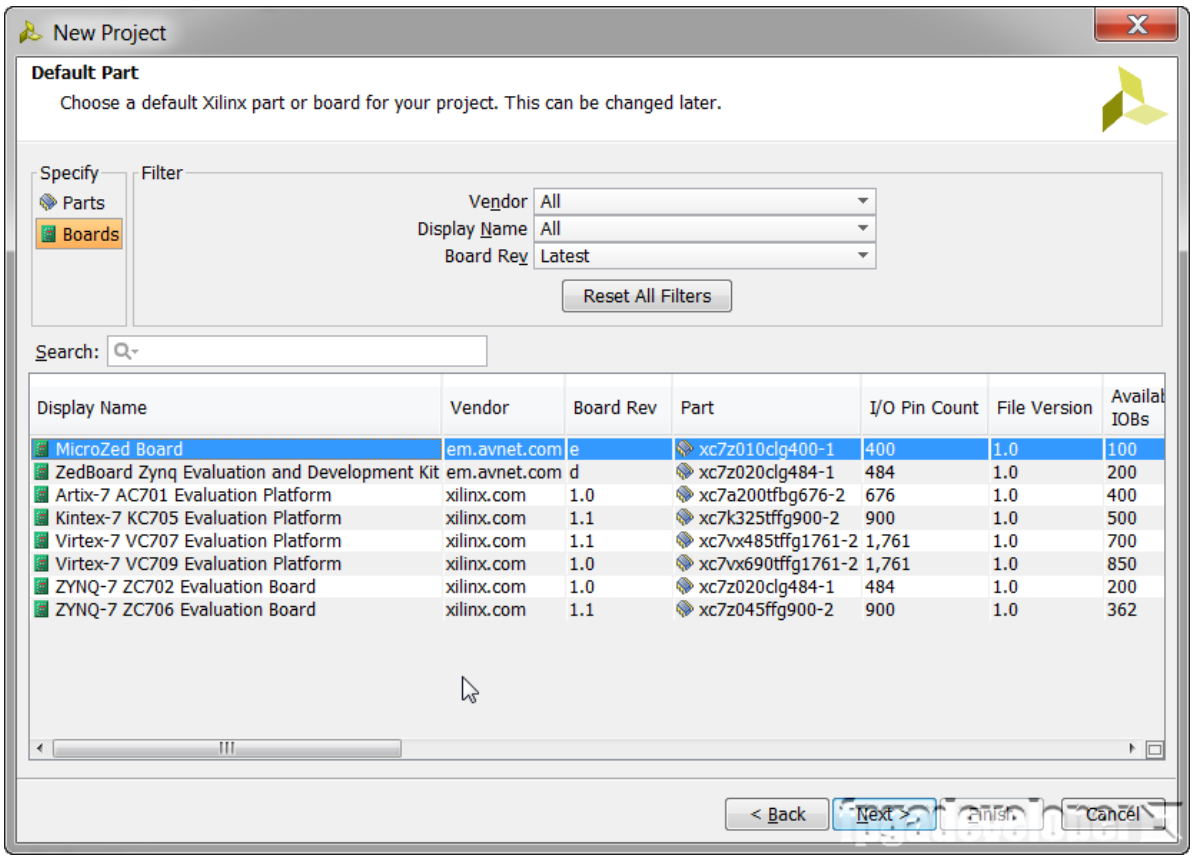


4. For the Add Sources window, click “Next”. We will add our multiplier source code later.

5. For the Add Existing IP window, click “Next”.

6. For the Add Constraints window, click “Next”.

7. For the Default Part window, select the “MicroZed Board” and click “Next”.



8. Click “Finish” to complete the new project wizard.

Change the project’s default language

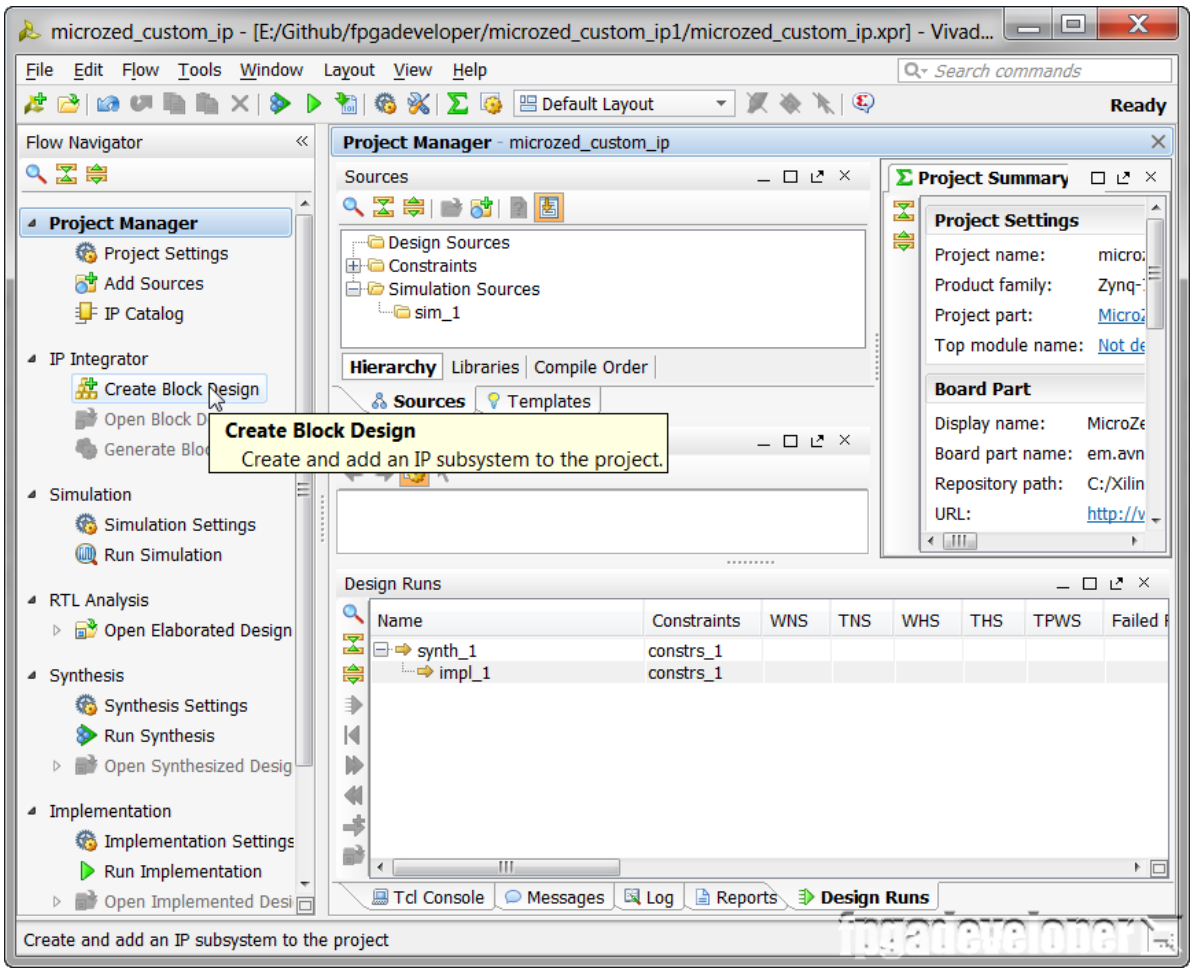
By default, the project will be created using Verilog as the default language. As we’ll be importing VHDL, let’s change that to VHDL:

1. From the menu, select Tools->Options.
2. In the “General” tab select target language : VHDL.

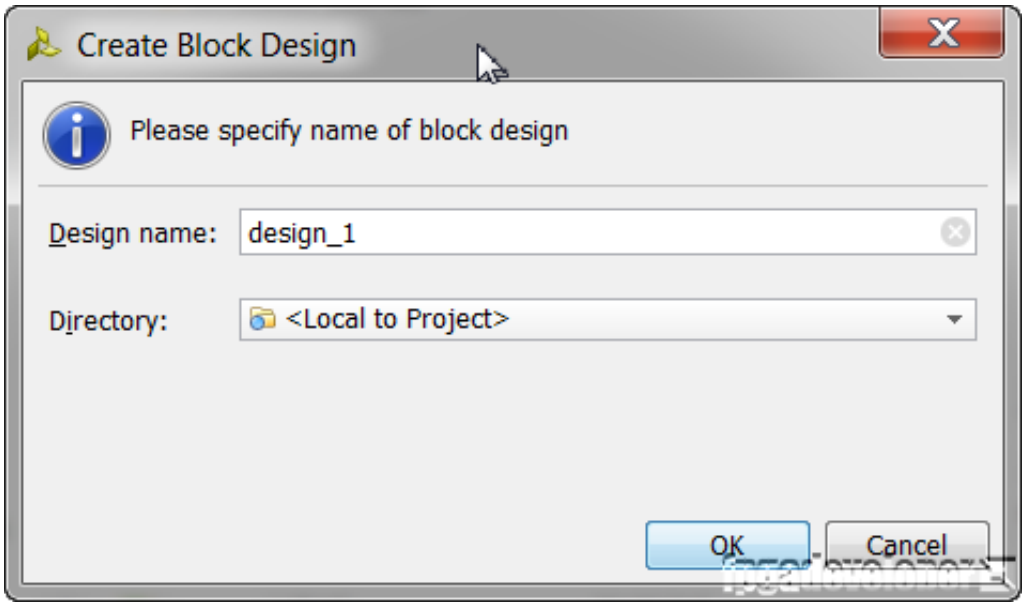
Setup the Zynq PS

The new Vivado project starts off blank, so to create a functional base design, we need to at least add the Zynq PS (processor system) and make the minimal required connections. Follow these steps to add the PS to the project:

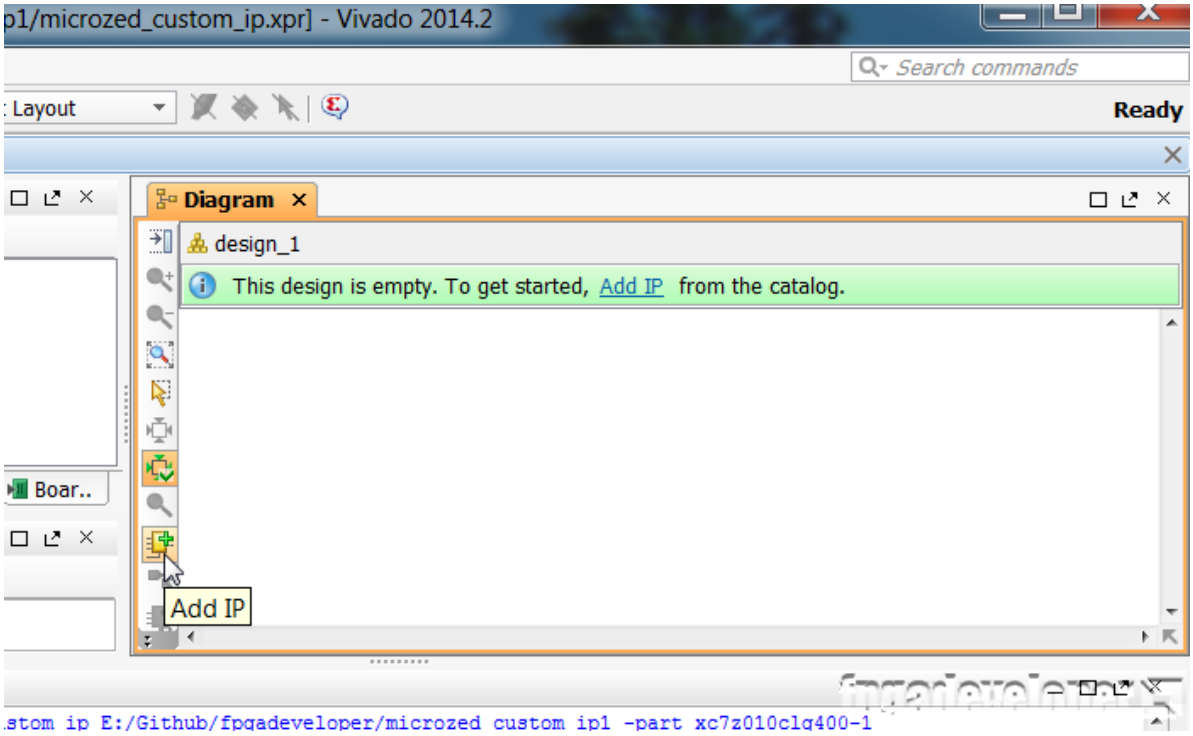
1. From the Vivado Flow Navigator, click “Create Block Design”.



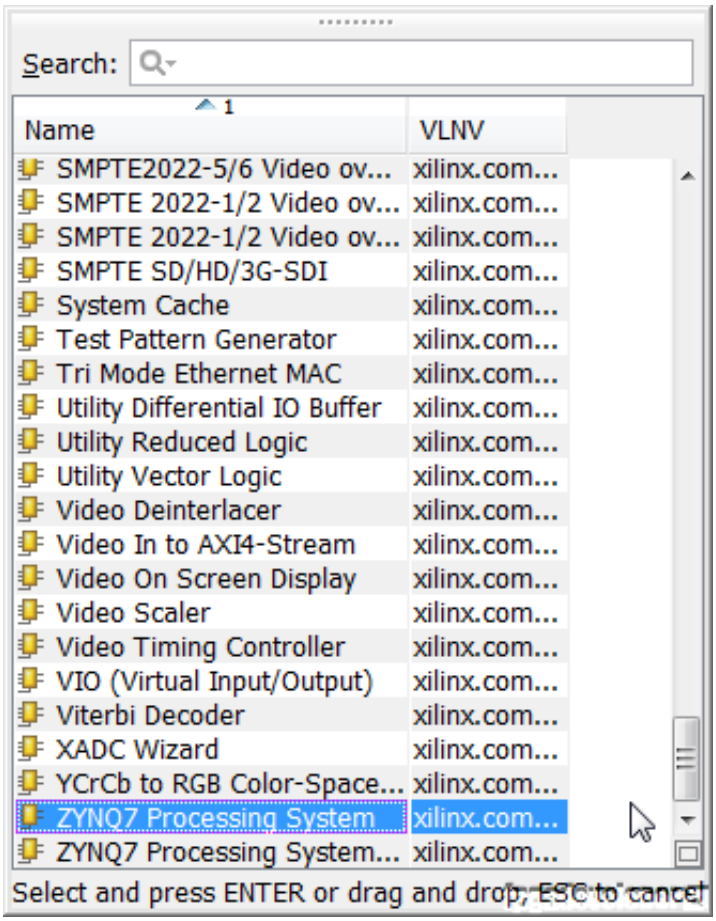
2. Specify a name for the block design. Let's go with the default "design_1" and leave it local to the project. Click "OK".



3. In the Block Design Diagram, you will see a message that says "This design is empty. To get started, Add IP from the catalog.". Follow this advice by clicking on the blue "Add IP" link, or by using the "Add IP" icon.

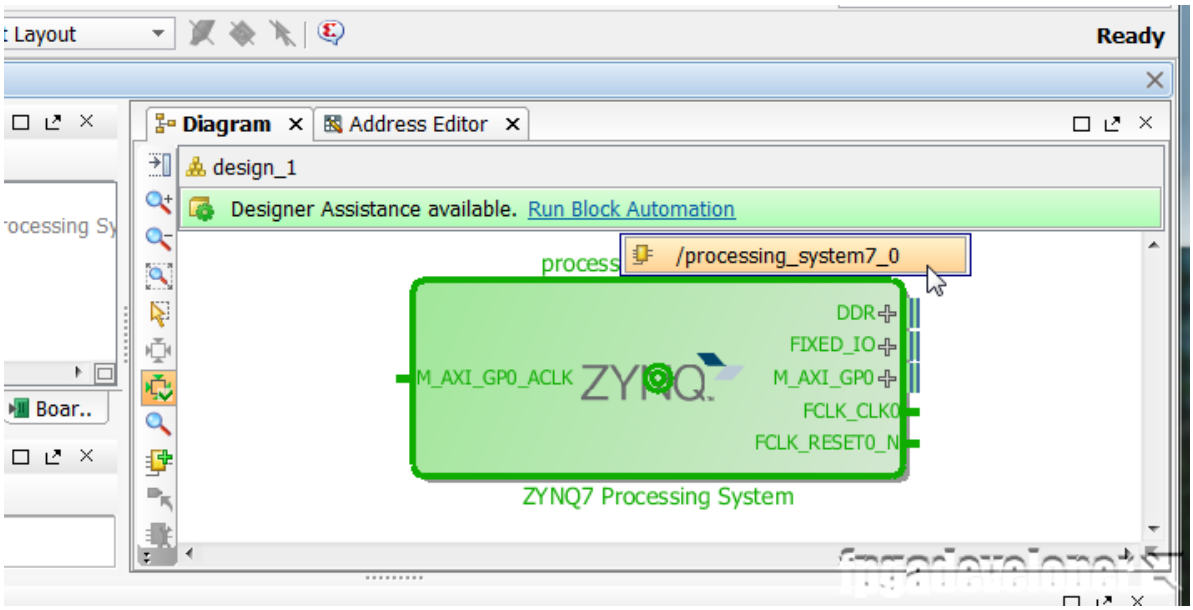


4. The IP catalog should appear. Go to the end of the list and double click on the block named "ZYNQ7 Processing System" – it should be the second last on the list. Vivado will now add the PS to the block diagram.

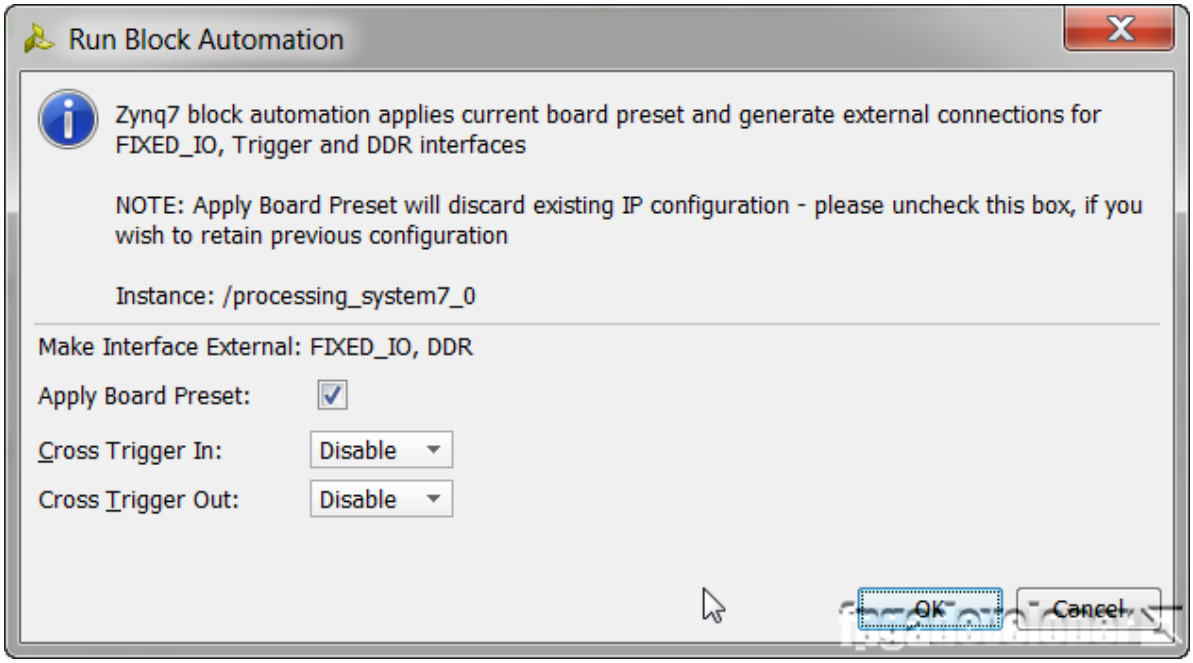


5. In the Block Design Diagram, you will see a message that says "Designer Assistance available. Run Block Automation". Click on the "Run Block Automation" link and select

"processing_system7_0" from the drop-down menu. Block Automation makes connections and pin assignments to external hardware such as the DDR and fixed IO. It does this using the board definition of the hardware platform you specified when you created the project (MicroZed). We could make these connections ourselves if we were using a custom board, but for off-the-shelf boards, Block Automation makes the process a lot easier.

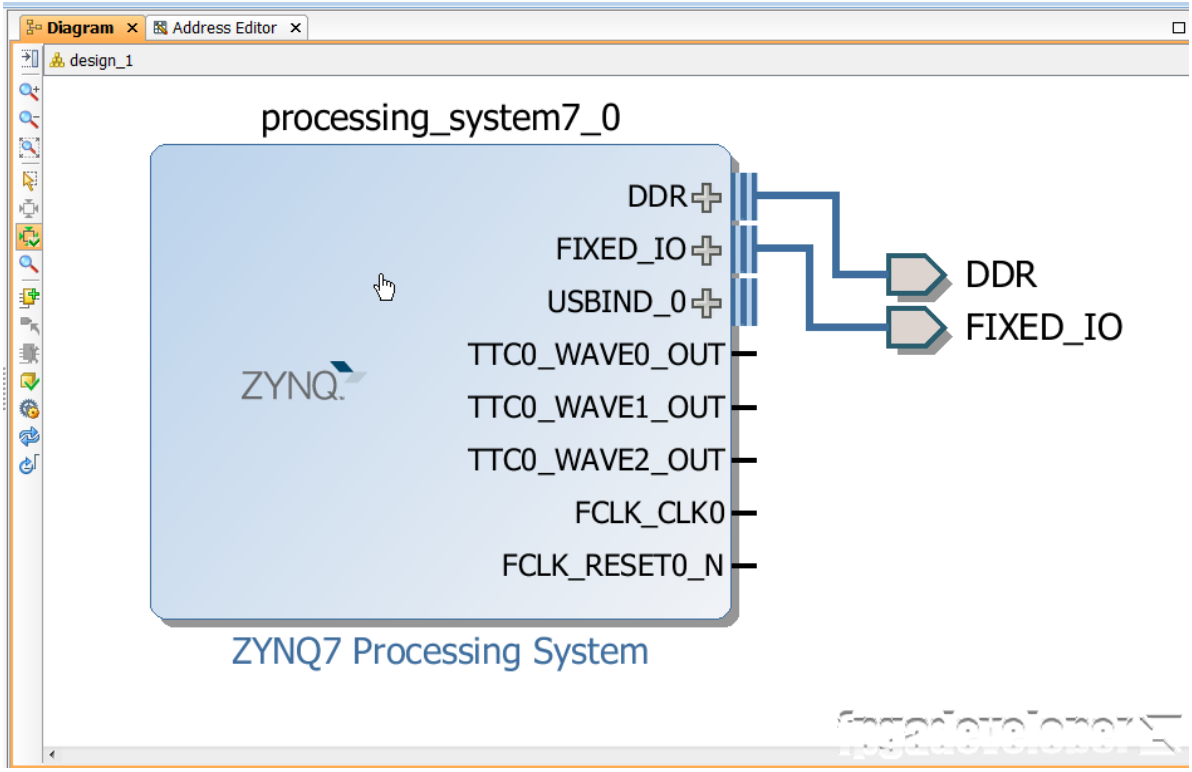


6. In the Block Automation window, make sure "Apply Board Preset" is ticked and click "OK".

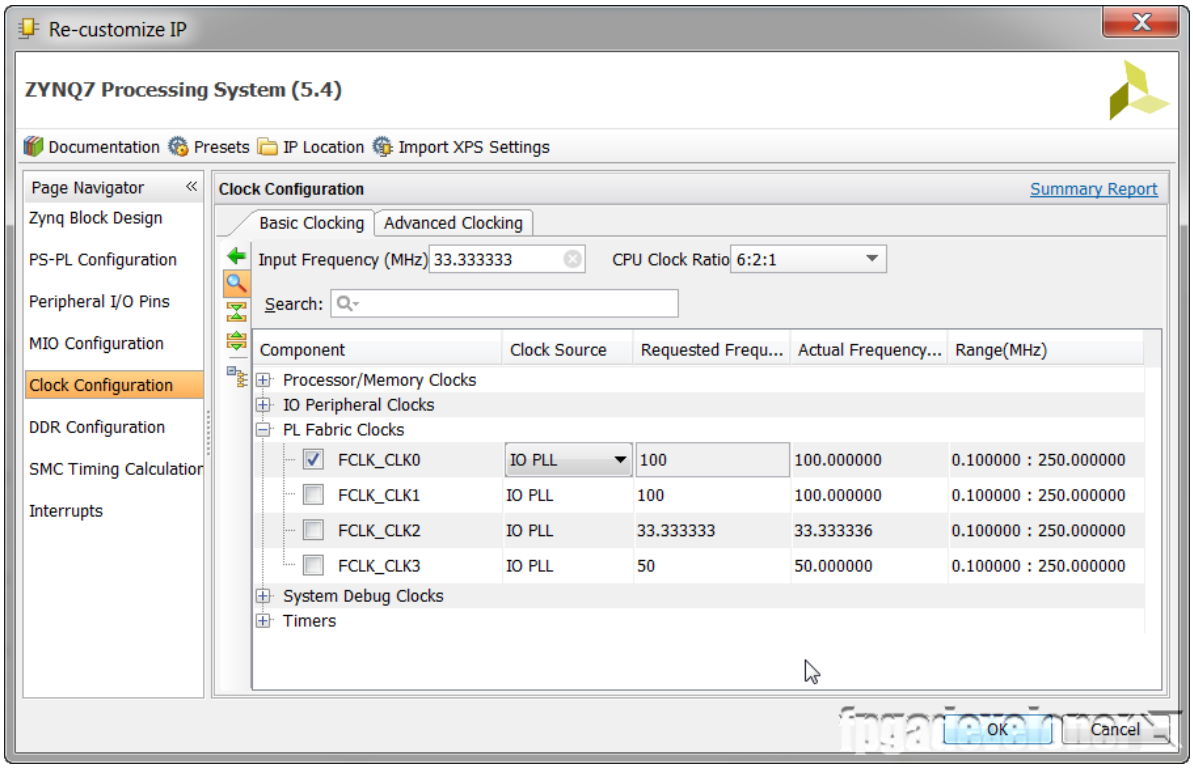


7. Now our block diagram has changed and we can see that the DDR and FIXED_IO are connected externally. Now the only remaining connection to make is the clock that we will use for the AXI buses. We must configure the Zynq to generate a clock and enable a general purpose AXI bus. To make these settings, double click on the

Zynq PS block.



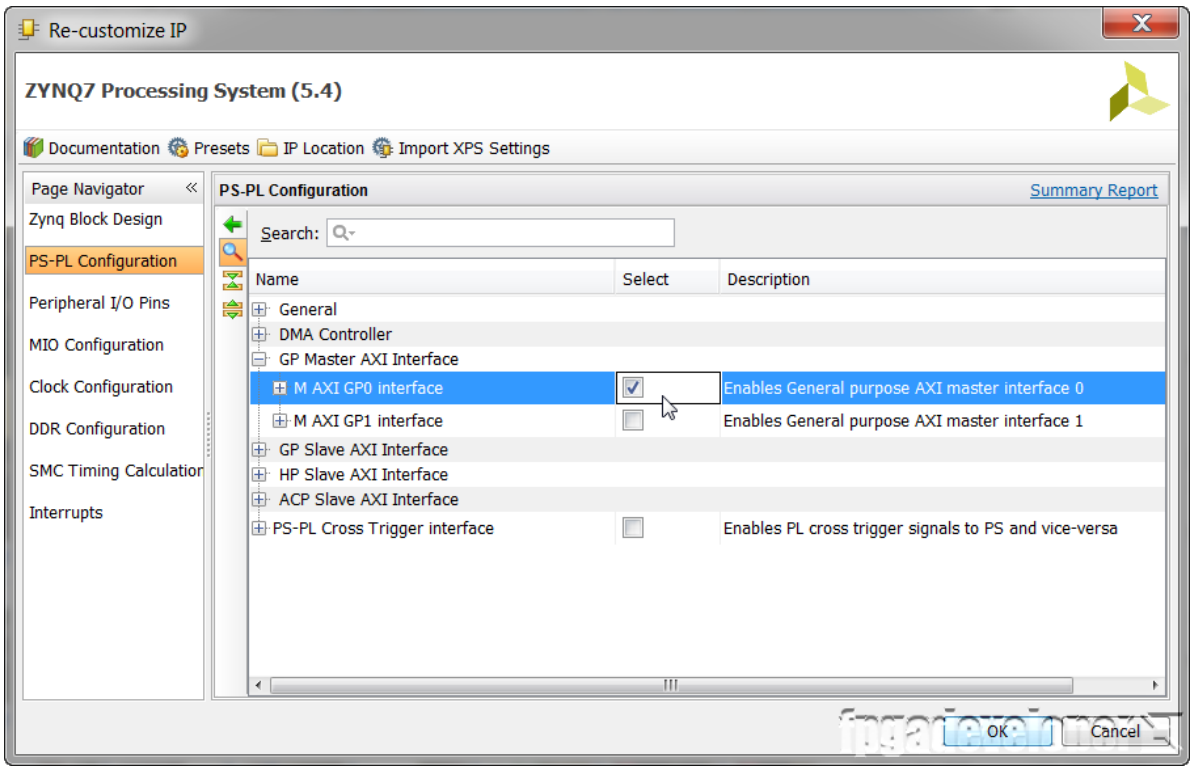
8. The Re-customize IP window will open. From the Page Navigator, select “Clock Configuration” and open the “PL Fabric Clocks” tree.



9. Make sure that the FCLK_CLK0 is enabled (ticked) and that it is set for a frequency of 100MHz. This will be our AXI clock.

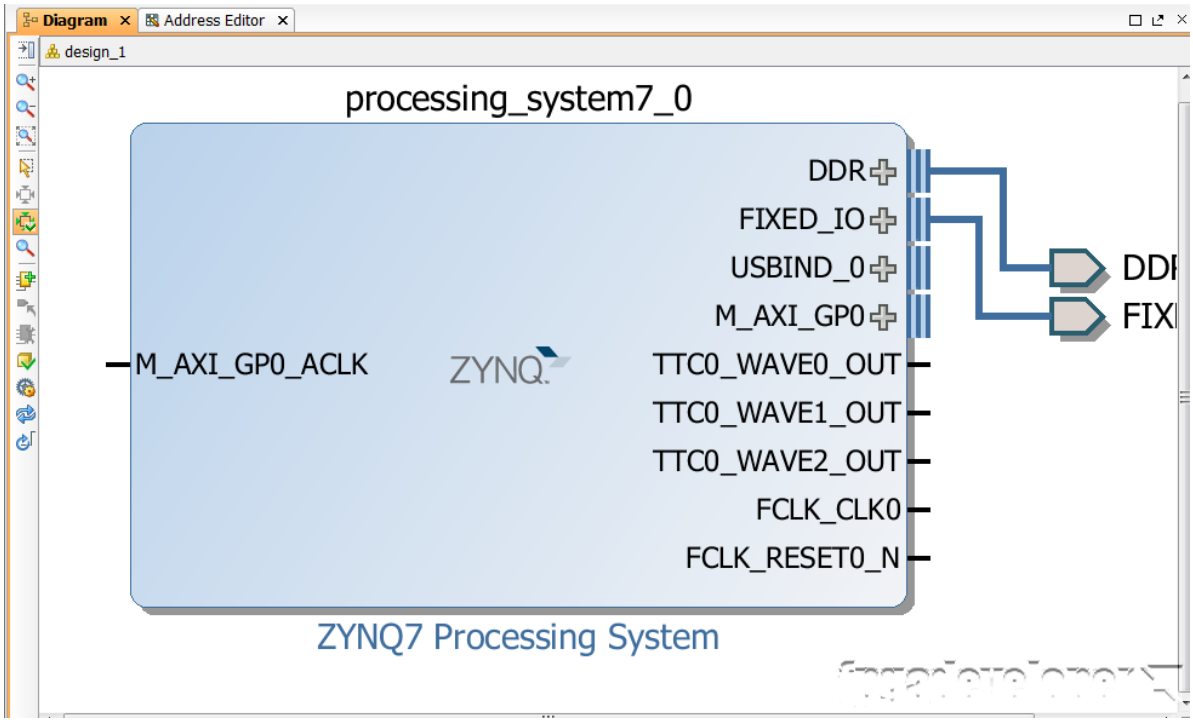
10. Now from the Page Navigator, select “PS-PL Configuration” and open the “GP Master AXI Interface” tree.

11. Tick the “M AXI GP0 interface” checkbox to enable it.

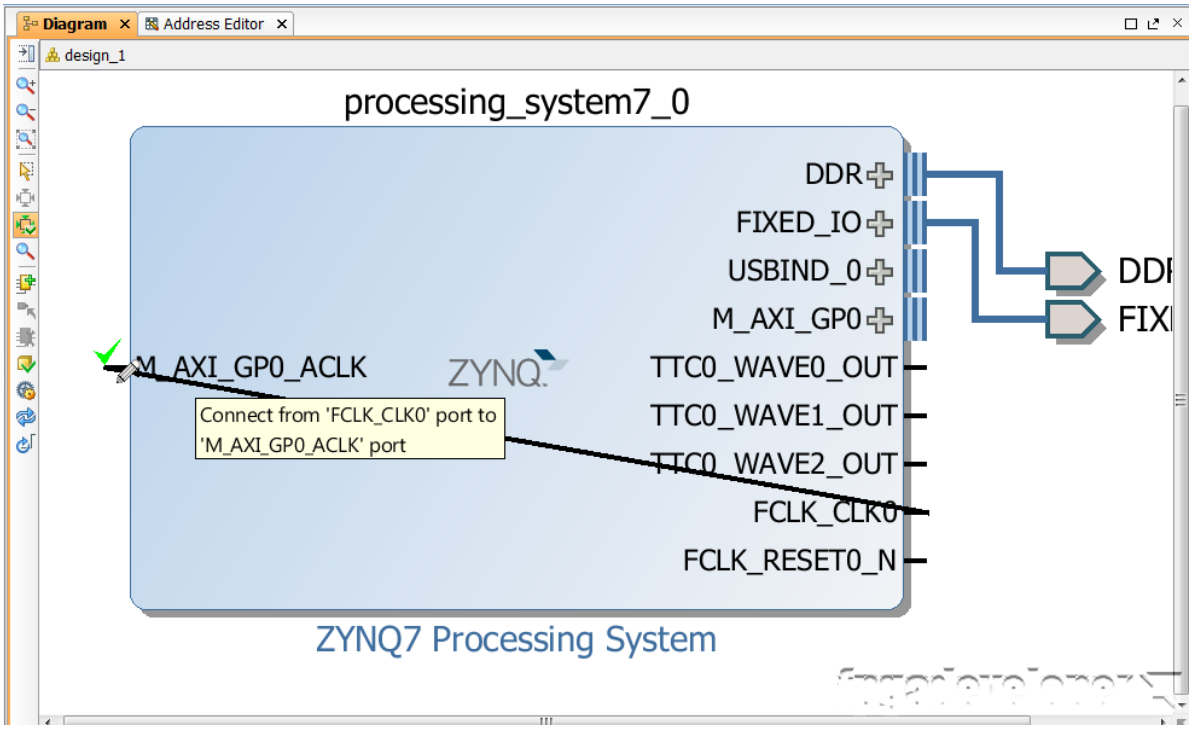


12. Now click “OK” to close the Re-customize IP window.

13. You should now see a new input port on the left side of the Zynq PS block. This is the AXI clock input.



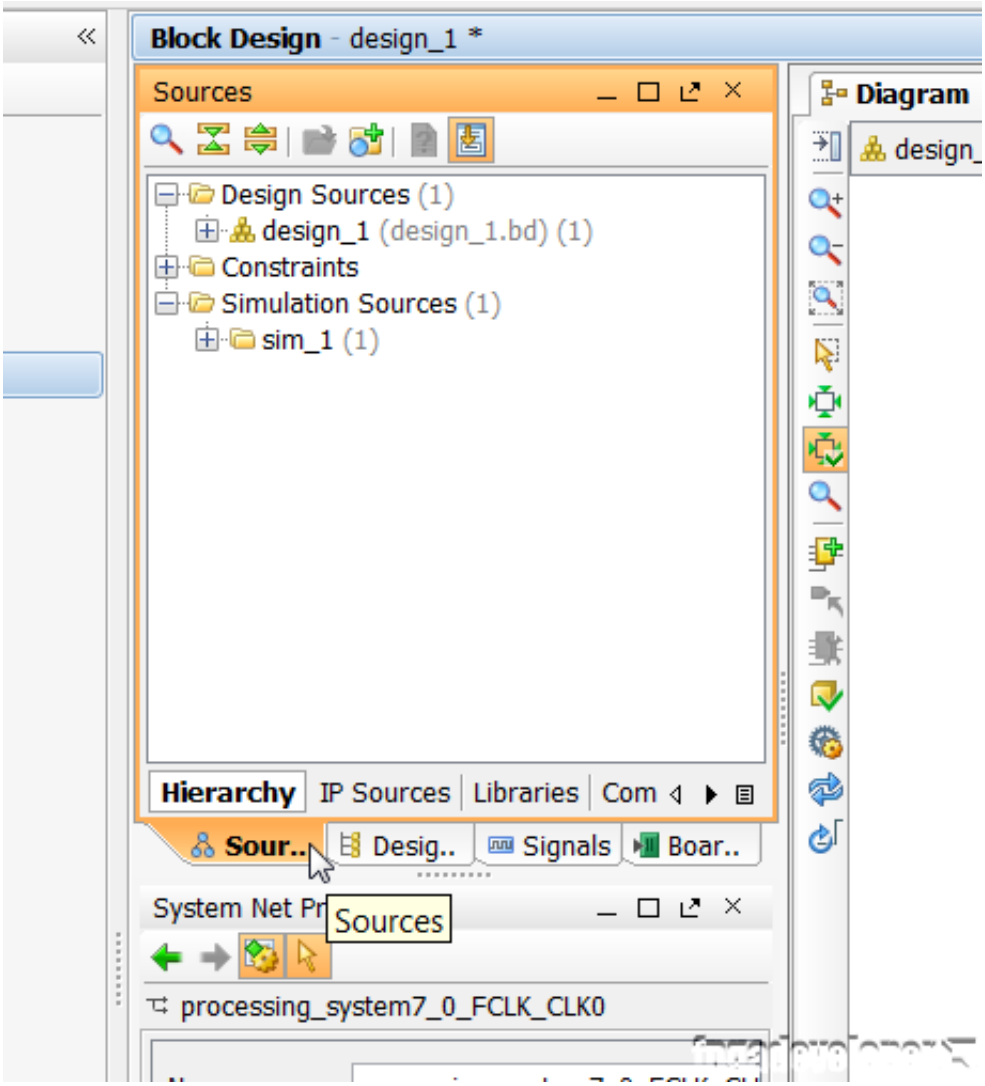
We must now connect the FCLK_CLK0 output to the AXI clock input. To do this, click on the FCLK_CLK0 output and then click on the M_AXI_GP0_ACLK input. This will trace a wire between the pins and make the connection.



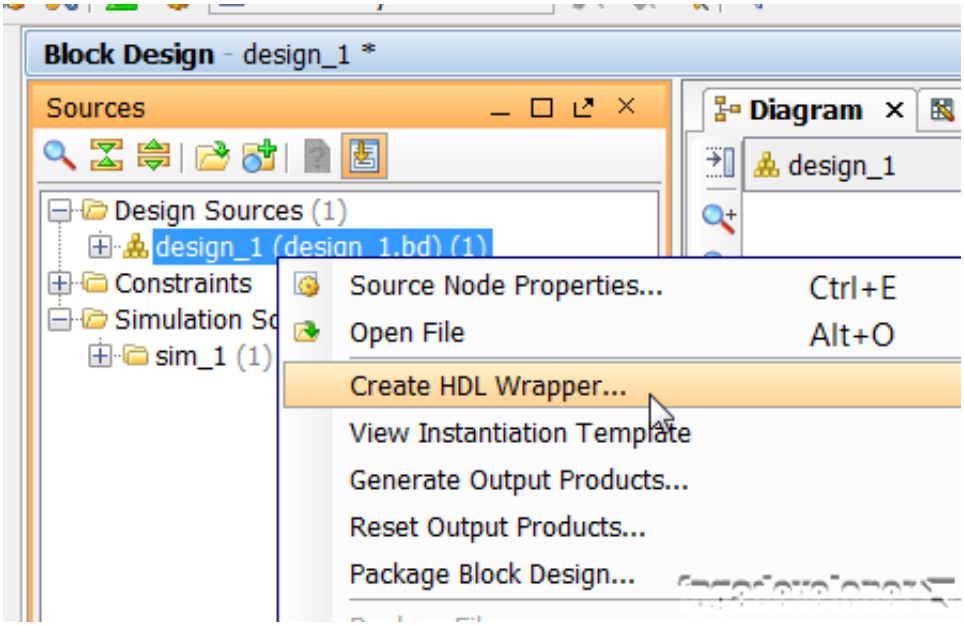
Create the HDL wrapper

Now the Zynq is setup and all we need to do to create a functional project is to create a HDL wrapper for the design.

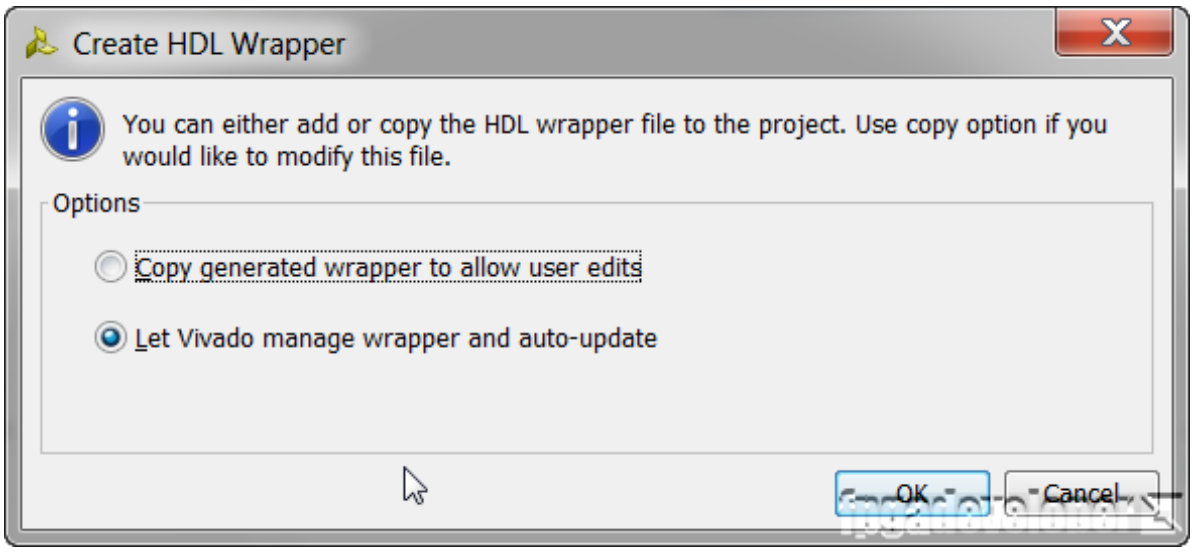
1. Open the “Sources” tab from the Block Design window.



2. Right-click on “design_1” and select “Create HDL wrapper” from the drop-down menu.



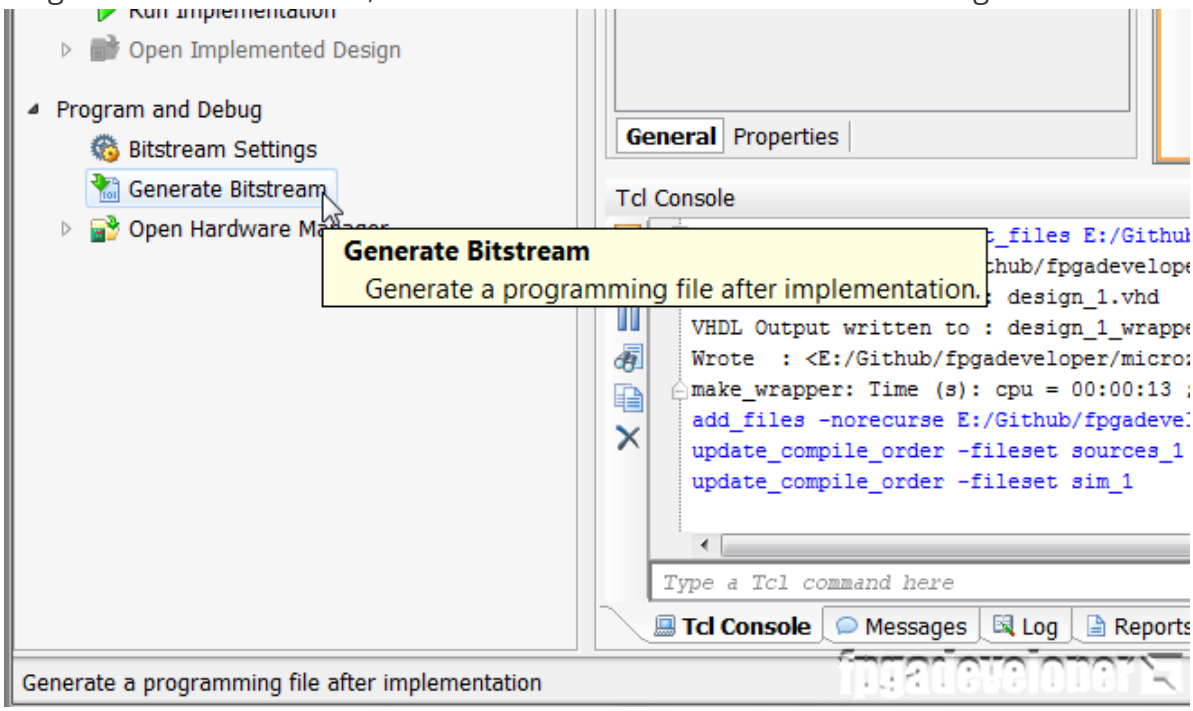
3. From the “Create HDL wrapper” window, select “Let Vivado manage wrapper and auto-update”. Click “OK”.



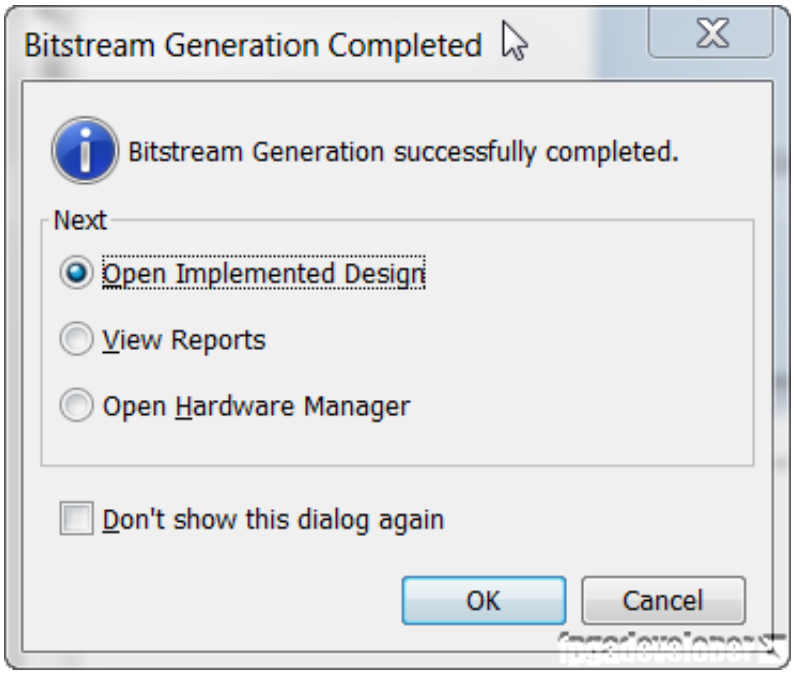
From this point, we have a base design containing the Zynq PS from which we could generate a bitstream and test on the MicroZed. We haven't exploited any of the FPGA fabric, but the Zynq PS is already connected to the Gigabit Ethernet PHY, the USB PHY, the SD card, the UART port and the GPIO, all thanks to the Block Automation feature. So there is already quite a lot we could do with the design at this point, such as running Linux on the PS or running a bare metal application on it.

Generate the bitstream

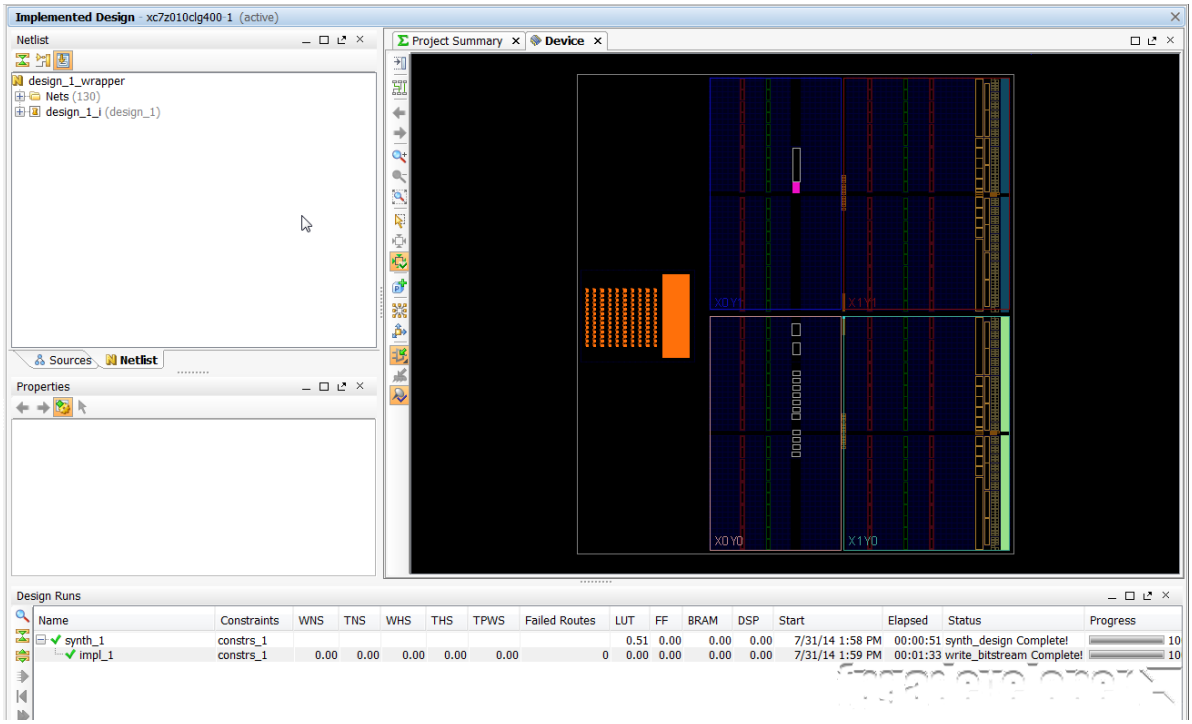
To generate the bitstream, click "Generate Bitstream" in the Flow Navigator.



Once the bitstream is generated, the following window appears. Select "Open Implemented Design" and click "OK".



The implemented design will open in Vivado showing you a map of the Zynq device and how the design has been placed. In our case, we haven't used any of the FPGA fabric (only the PS), so the map is empty for the most part.



~~Export the hardware to SDK~~

~~Once the bitstream has been generated, the hardware design is done and we're ready to develop the code to run on the processor. This part of the design process is done in Xilinx Software Development Kit (SDK), so from Vivado we must first export the project to SDK.~~