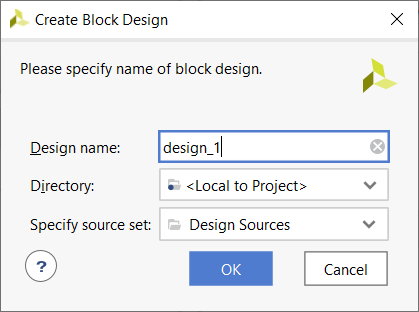
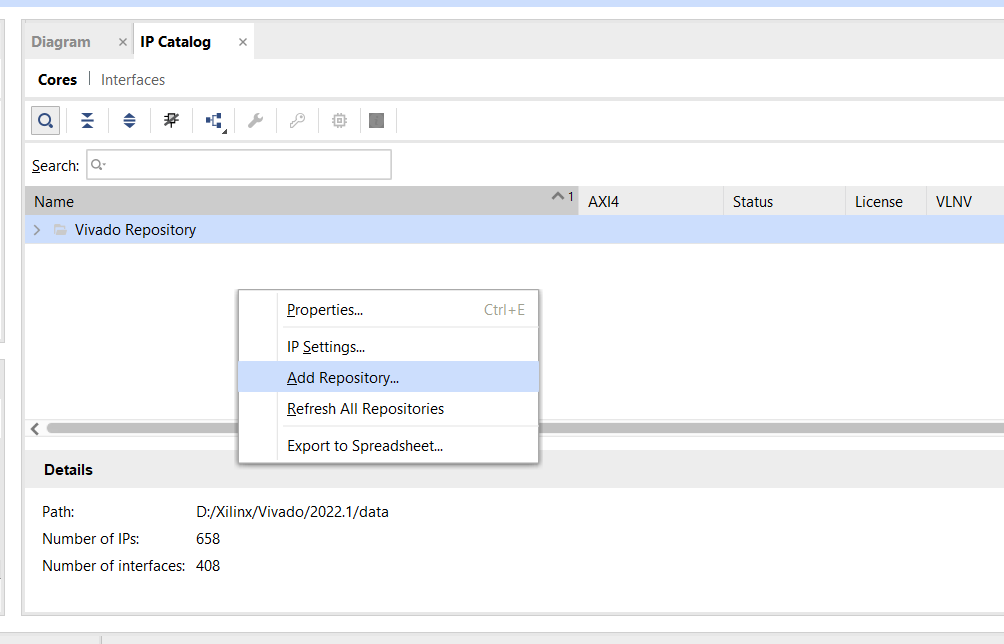
HowTo Interface IP to the Zynq

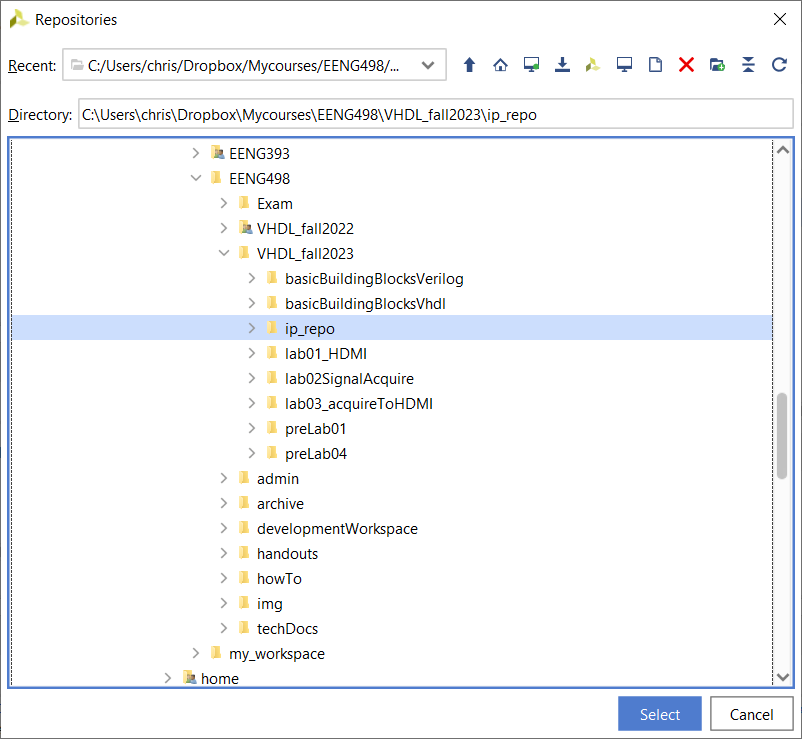
I created a new project called ZynqWithPwm. In the Flow Navigator click on Create Block Diagram. In the Create Block Design pop-up, provide a design name. Feel free to give your design a better name than design\_1, I used ZynqQWithPwm.



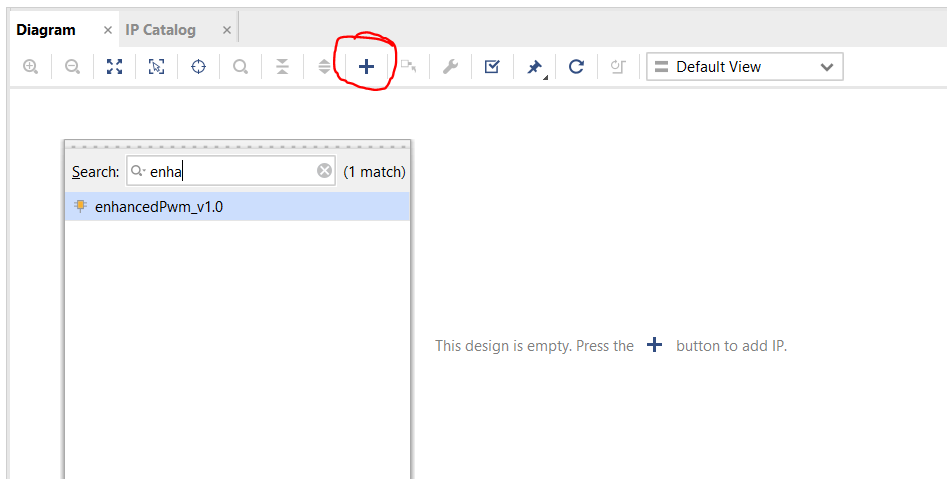
You will have a blank canvas to place parts on. Let’s start by adding our enhancedPwm. Click on the IP Catalog link in the Flow Navigator menu. Collapse the Vivado Repository and right click in the blank area and select “Add Repository…”



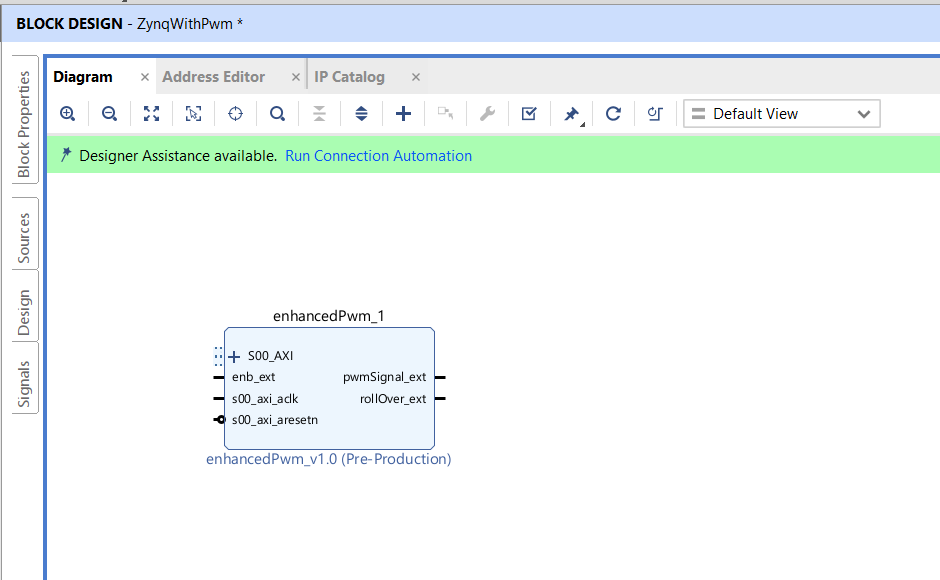
Navigate to the directory where you stored the enhancedPwm IP and click select.



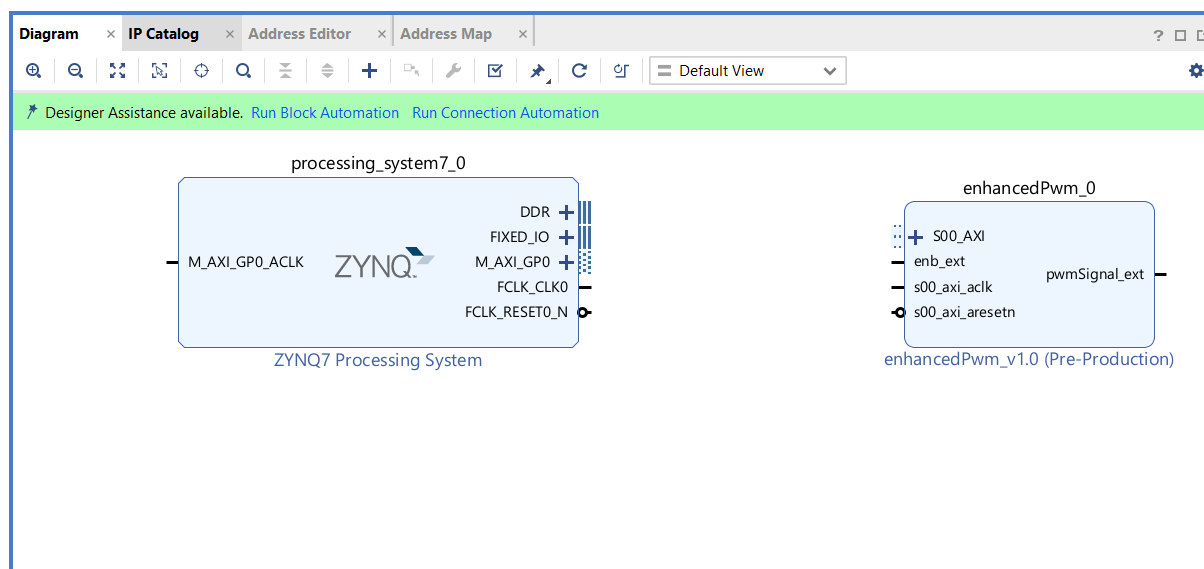
Now go back to the Diagram tab and click on the Add IP button (circled in red) in the Diagram tool bar. Type “enhan” in the search area and select enhancedPwm\_ip\_v1.0.



You should see the enhancedPwm component in the Diagram window.

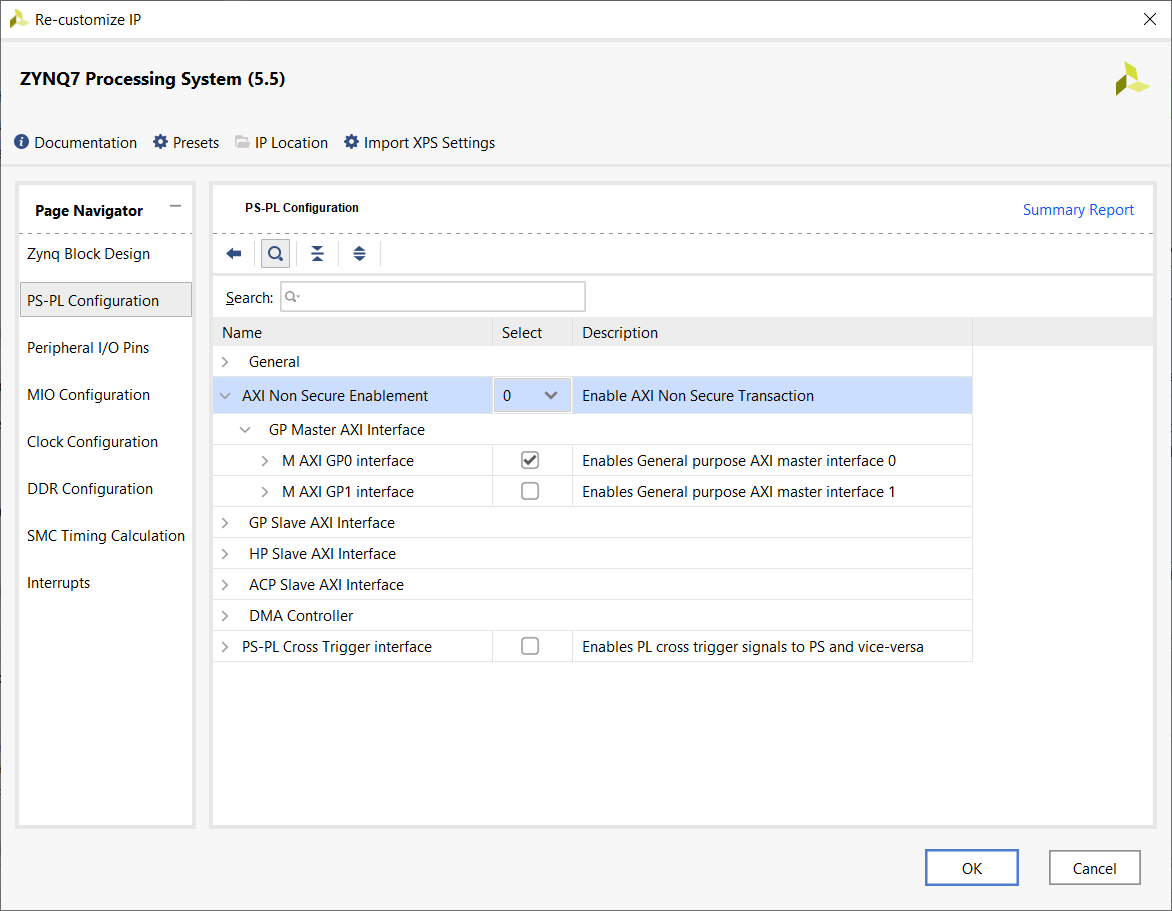


Again, click on the Add IP button in the Diagram tool bar. Type “zy” in the search area select ZYNQ7 Processing System. You now have the two main components of your design.



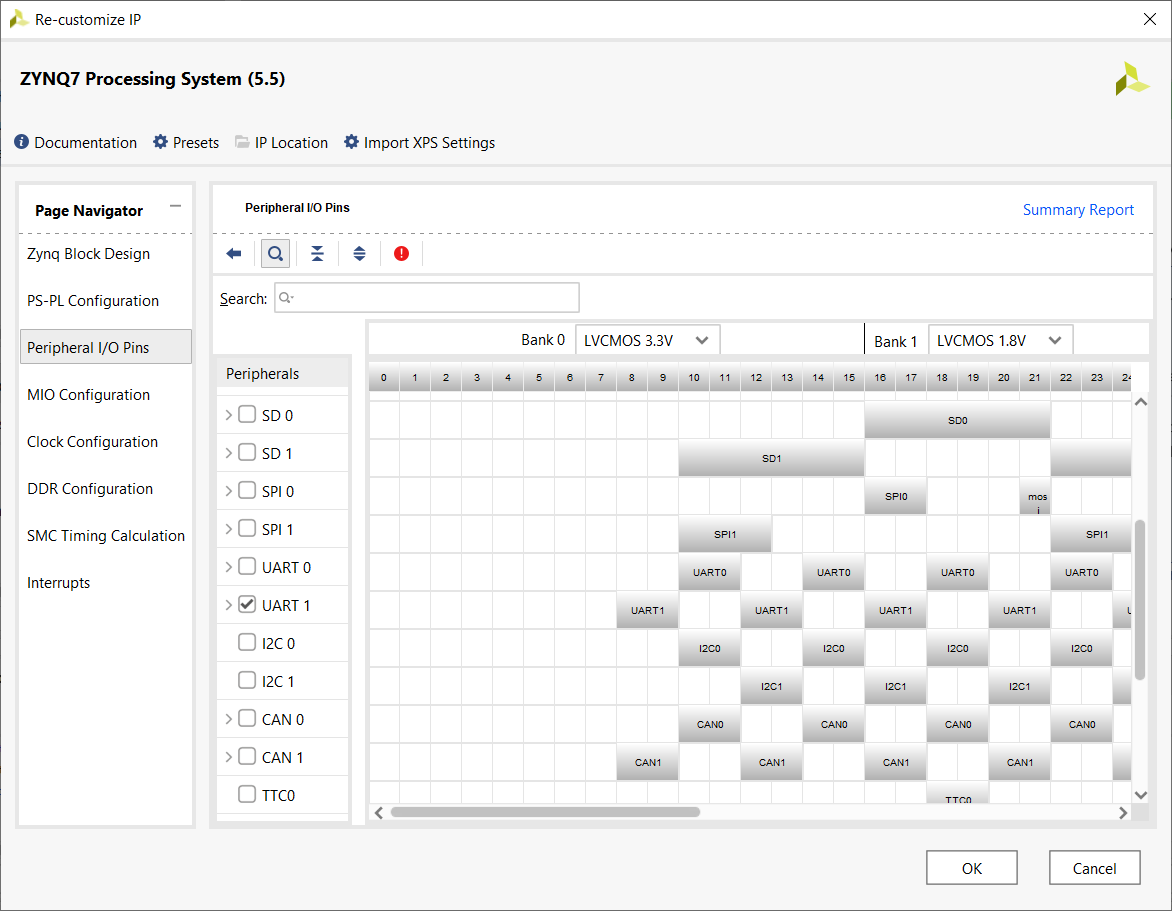
You now need to customize the Zynq processor and connect the processor to your custom IP. Start by customizing the Zynq processor by doubling clicking on the Zynq processing block. Go through the steps listed in the Page Navigator bar at left.

**PS-PL Configuration** – Leave default.

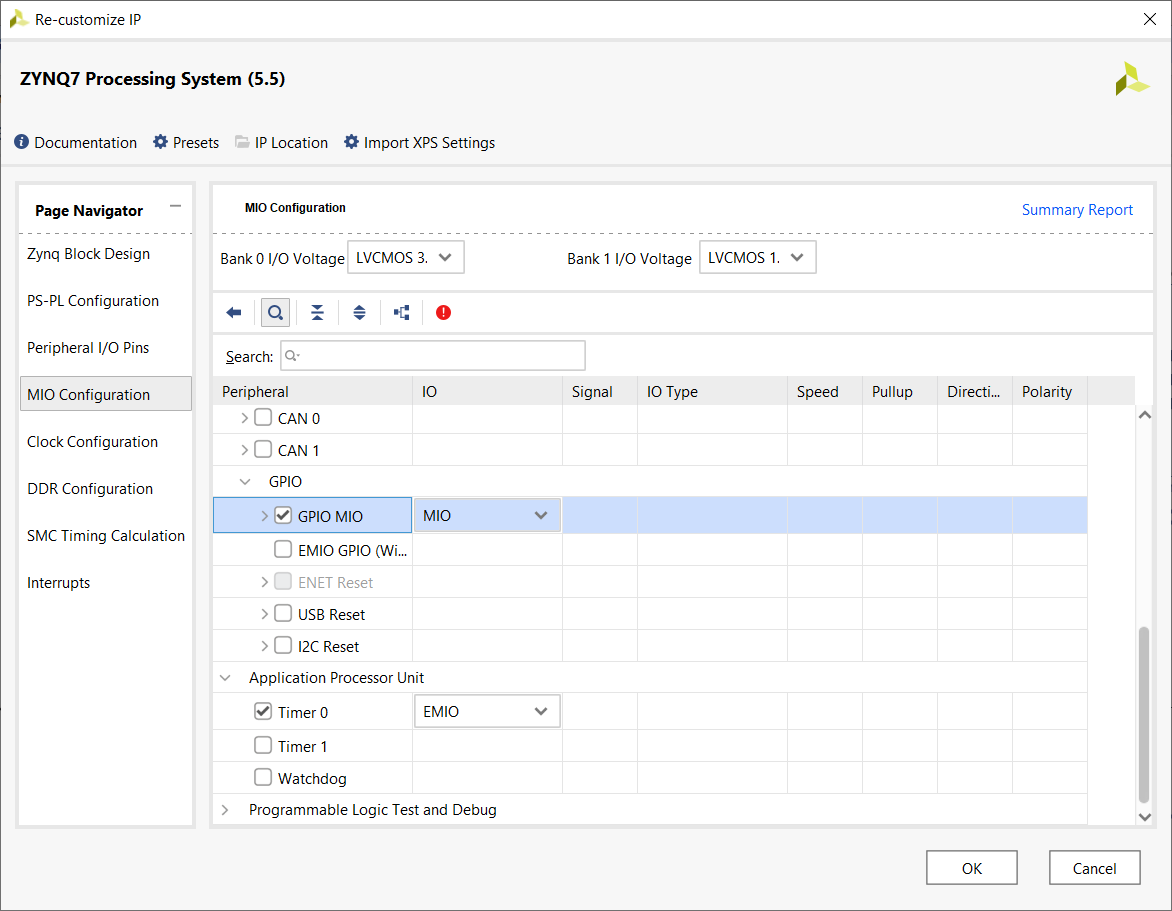


**Peripheral I/O Pins**

Select Bank 1 = LVCMOS 1.8V and check USART 1. If you scroll right, you will see pins 48,49 highlighted green for USART1. If you select HSTL1.8V, you will not be able to RX characters to the Zynq – ask me how I know :/



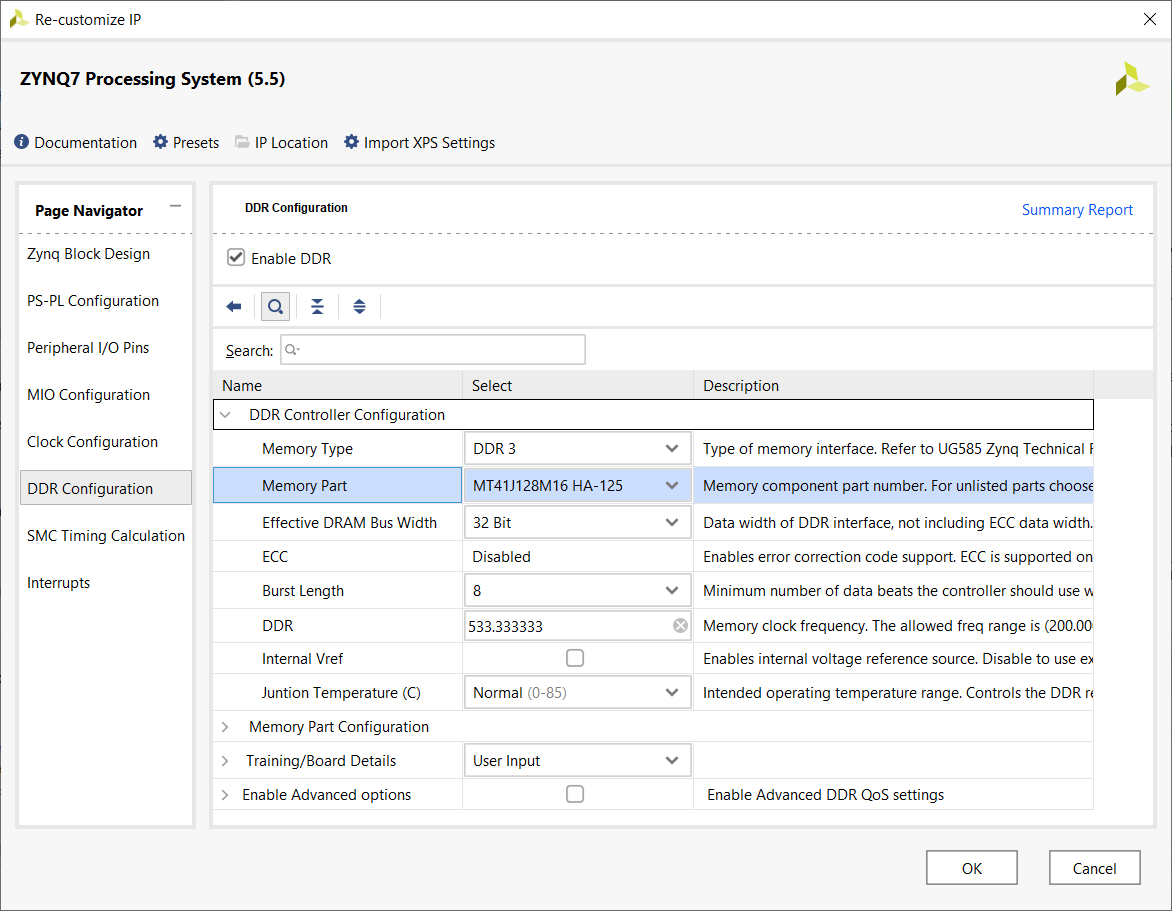
**MIO Configuration** – Enable GPIO MIIO and Timer 0 EMIO.



**Clock Configuration** – Leave defaults.

**DDR Configuration**

Memory Part: MT41J128M16 HA-125 Scroll down to find it, about half way.



**SMC Timing Calculations** – Leave defaults.

**Interrupts** – Enable Fabric Interrupts. Open the PL-PS Interrupts and check Core0\_nFIQ.

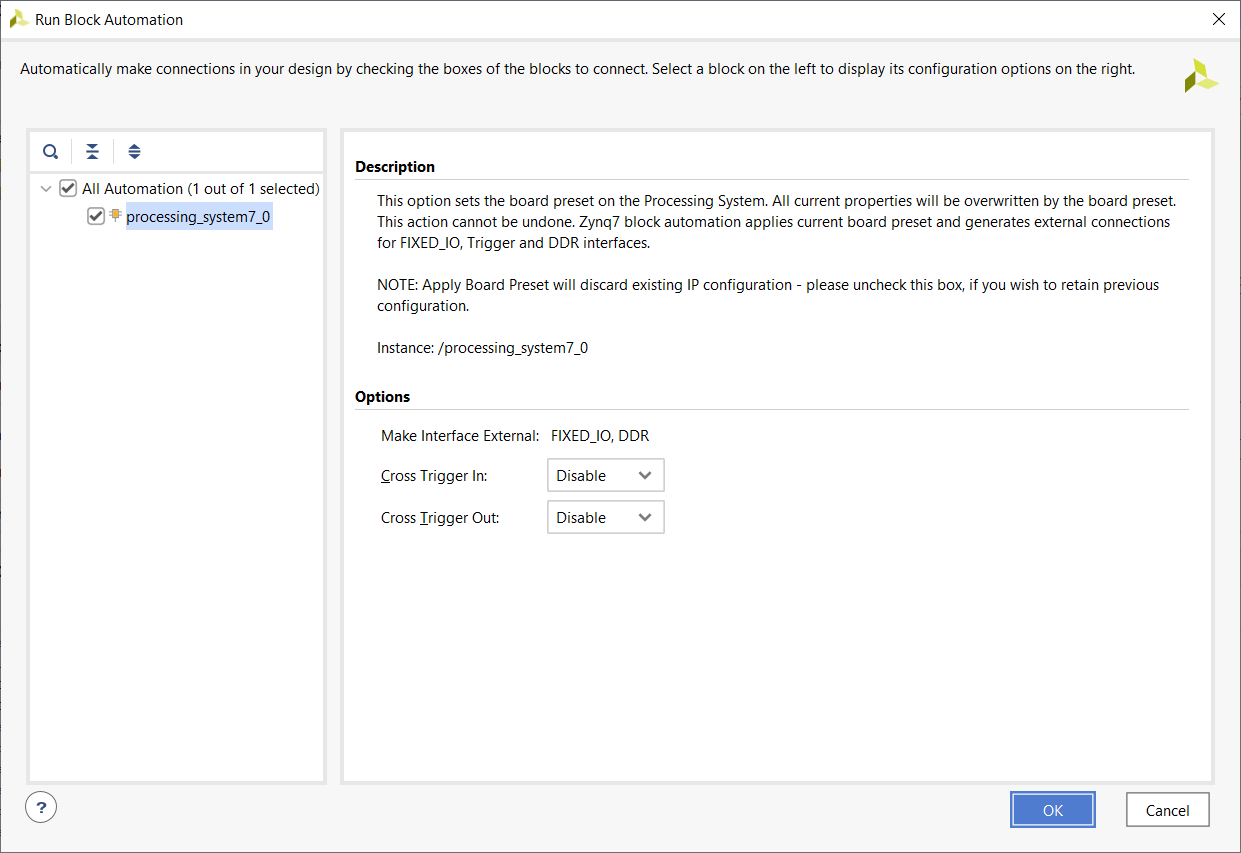
A screenshot of a computer

Description automatically generated

Finish configuration by clicking OK to close the Re-customize IP wizard.

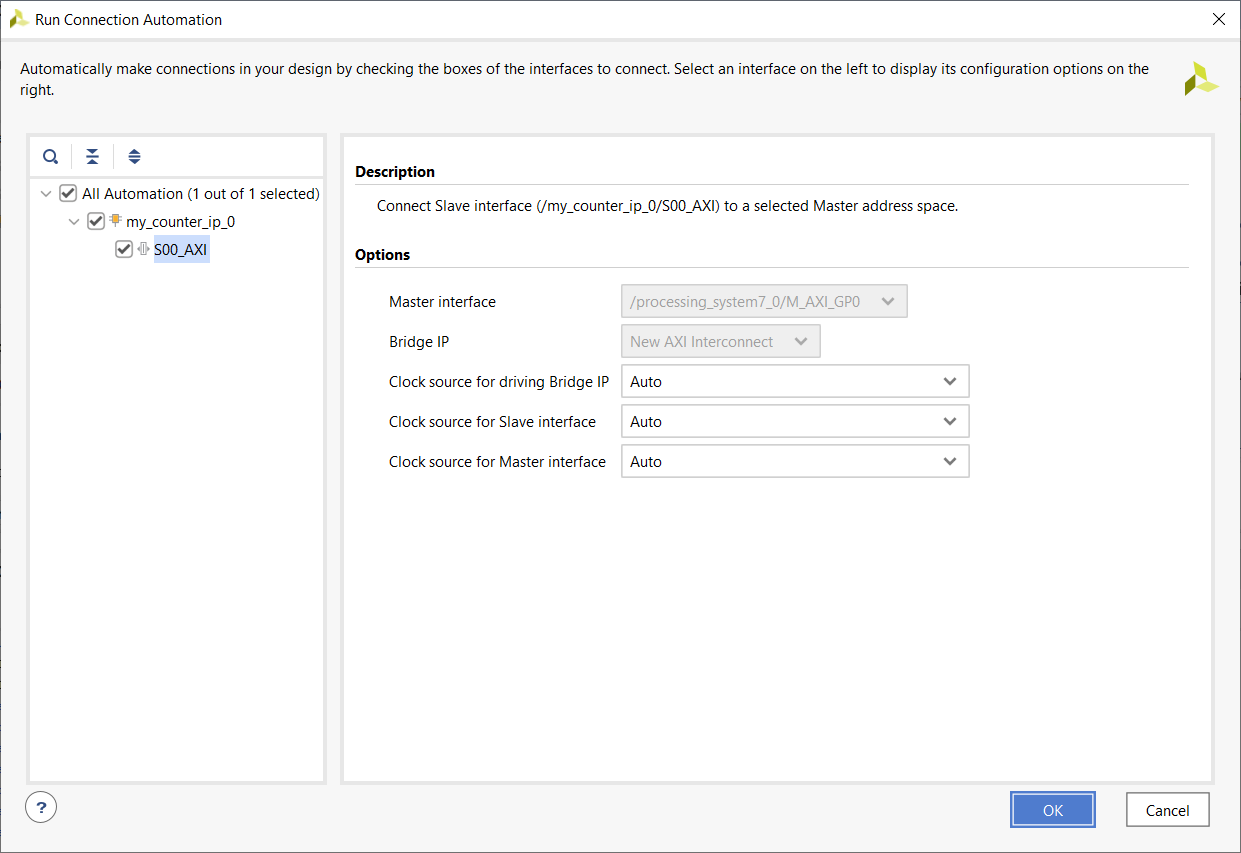
Connect Zynq processor to custom IP

In the block diagram window, Click Run Block Automation

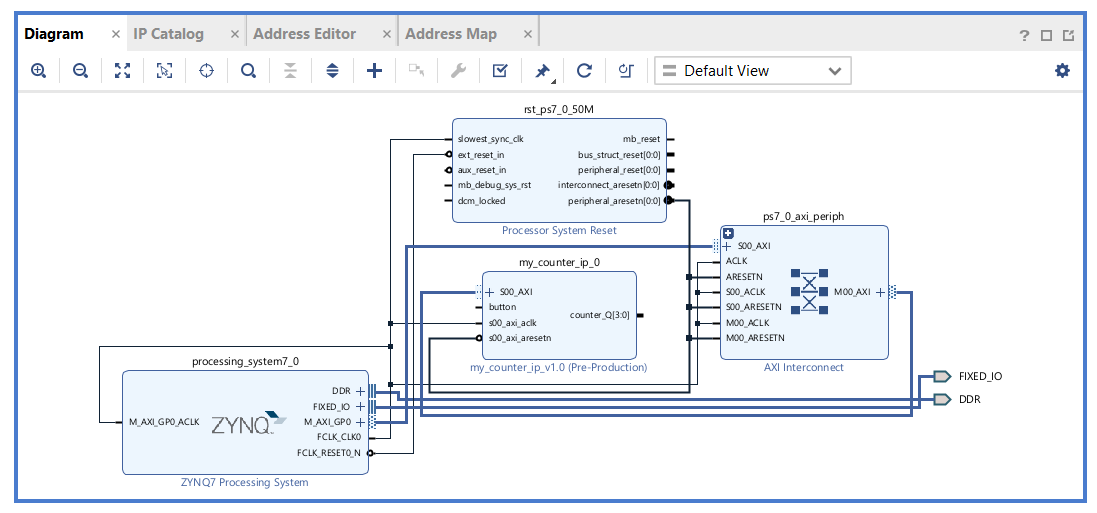


Leave defaults then click OK.

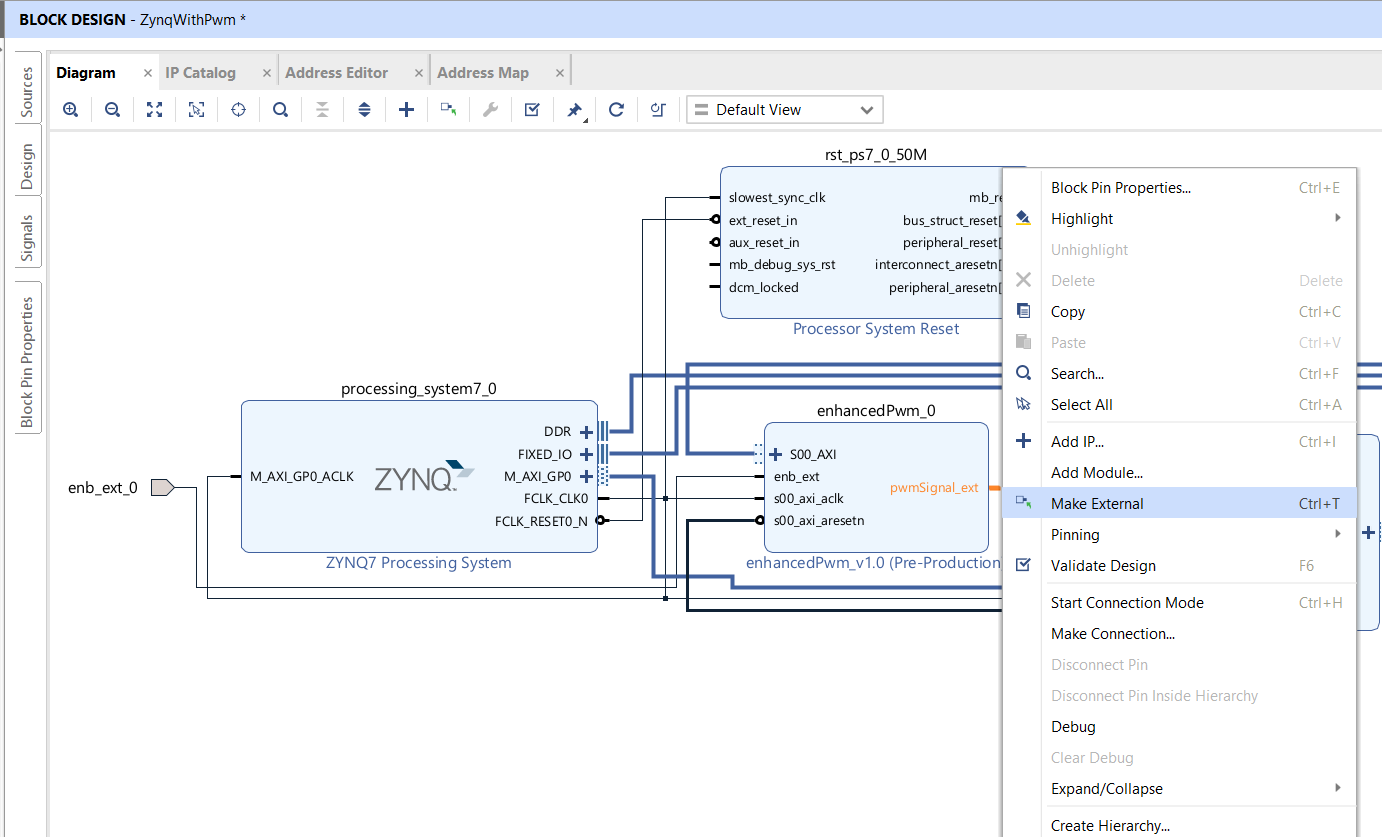
Back in the block diagram, Click Run Connection Automation.



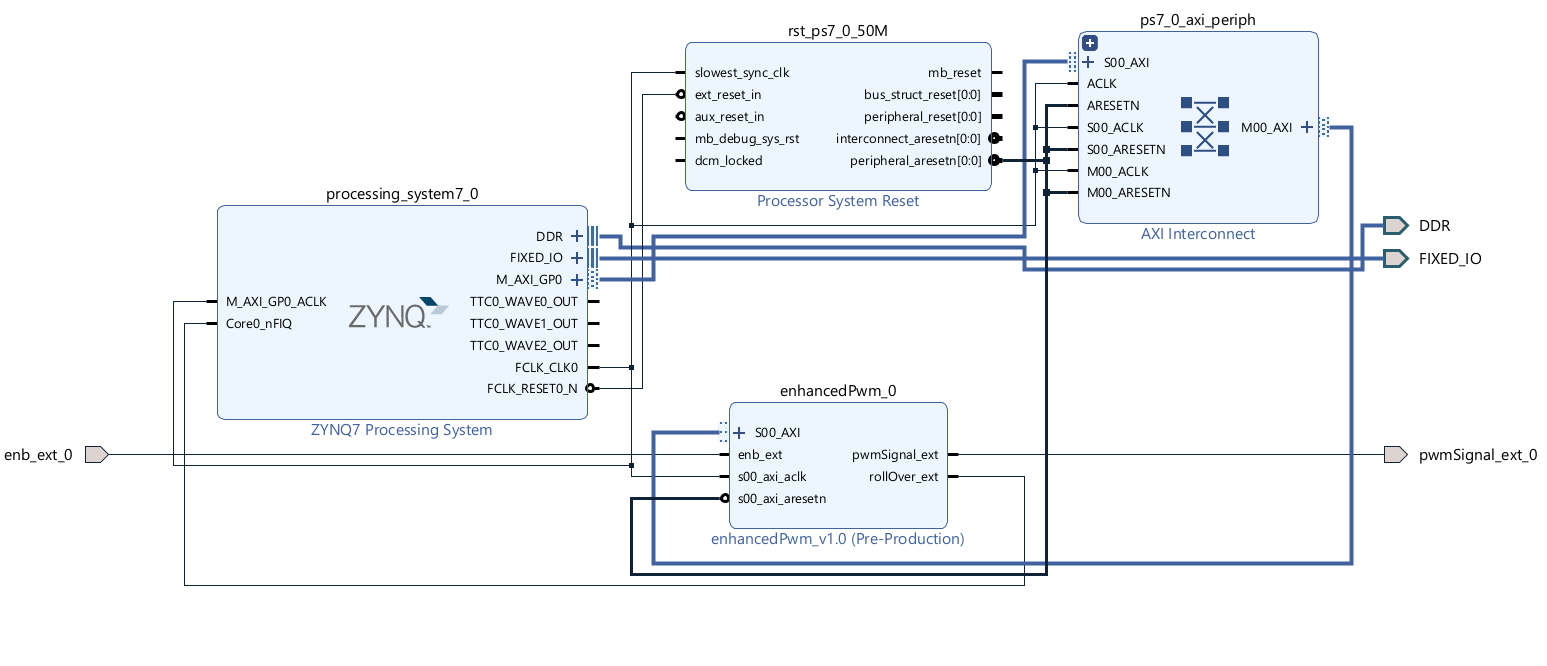
Leave defaults then click OK.



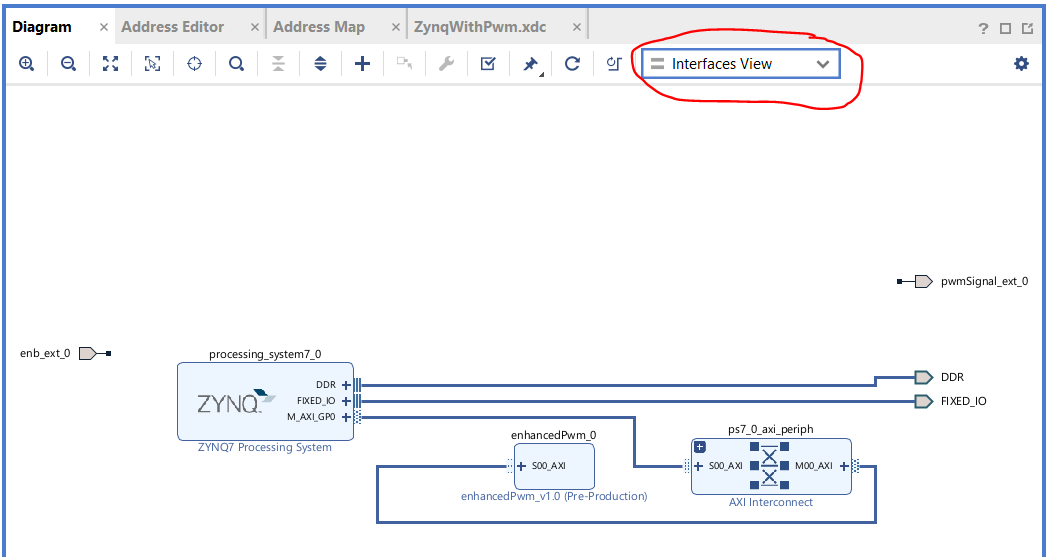
Connect the enb\_ext and pwmSignal\_ext on the enhancedPwm\_ip\_0 component to pins outside the Zynq chip. To do this, right mouse click on the enb\_ext on the enhancedPwm\_ip\_0 component and select Make External. Repeat for pwmSignal\_ext.



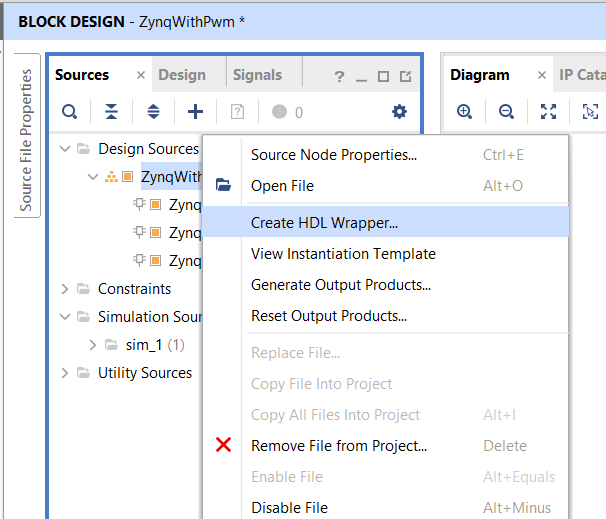
Connect the rollover\_ext signal on the enhancedPwm component to the Core0\_nFIQ input on the Zynq.



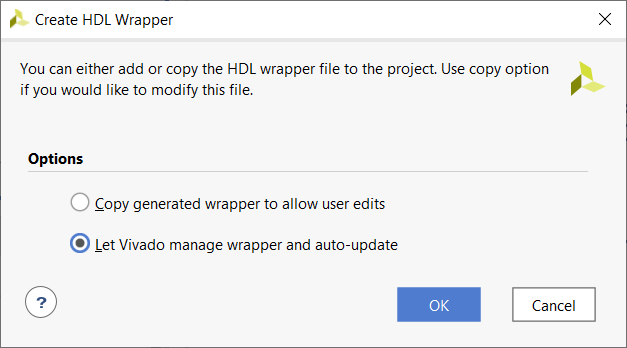
You can use the optimize routing tool to help clean-up the wiring. You may find that changing the default view provides a more streamlined and intuitive view of the project.



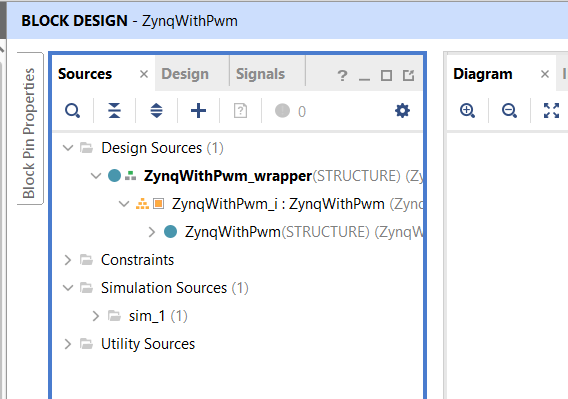
Next, create an HDL file for this design by right clicking on design\_1 and selecting Create HDL Wrapper.



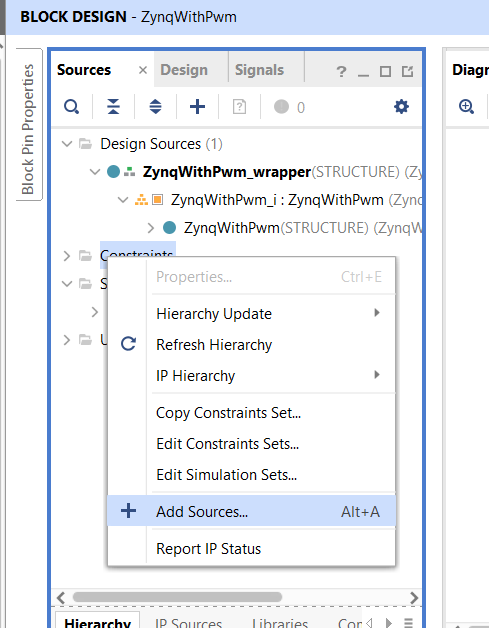
In the Create HDL Wrapper Wizard, leave the defaults and click OK.



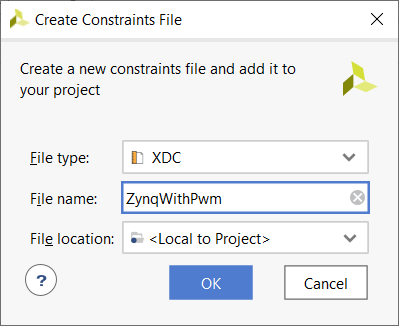
Your block diagram should now be wrapped in a VHDL file.



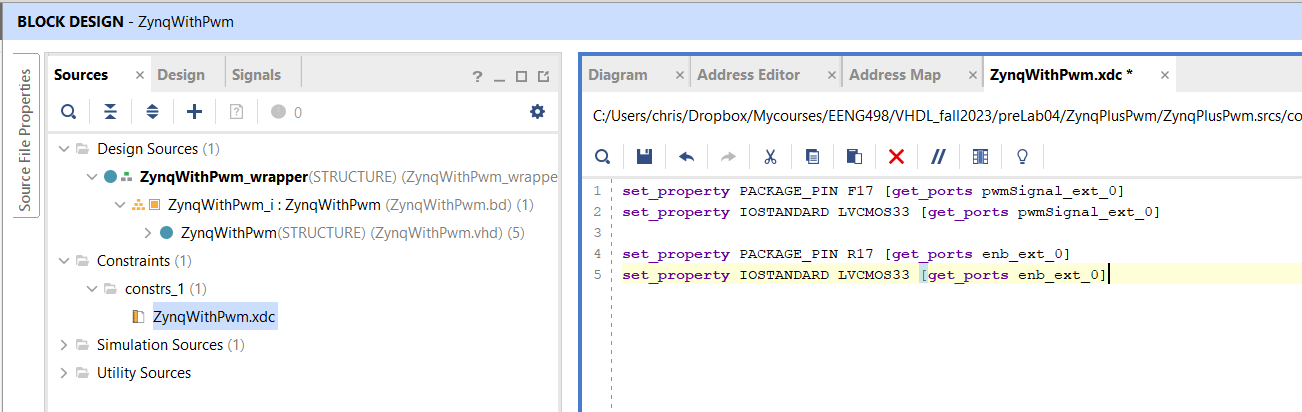
Next, let’s add a constraints file that provides the location for the enb\_ext and pwmSignal\_ext signals. Right mouse click on the constraints folder in the sources tab and select Add Sources…



Keep the default Add Or Create Constraints radio button select, click Next. Click Create File. In the pop-up, choose a suitable name, click OK, then Finish.

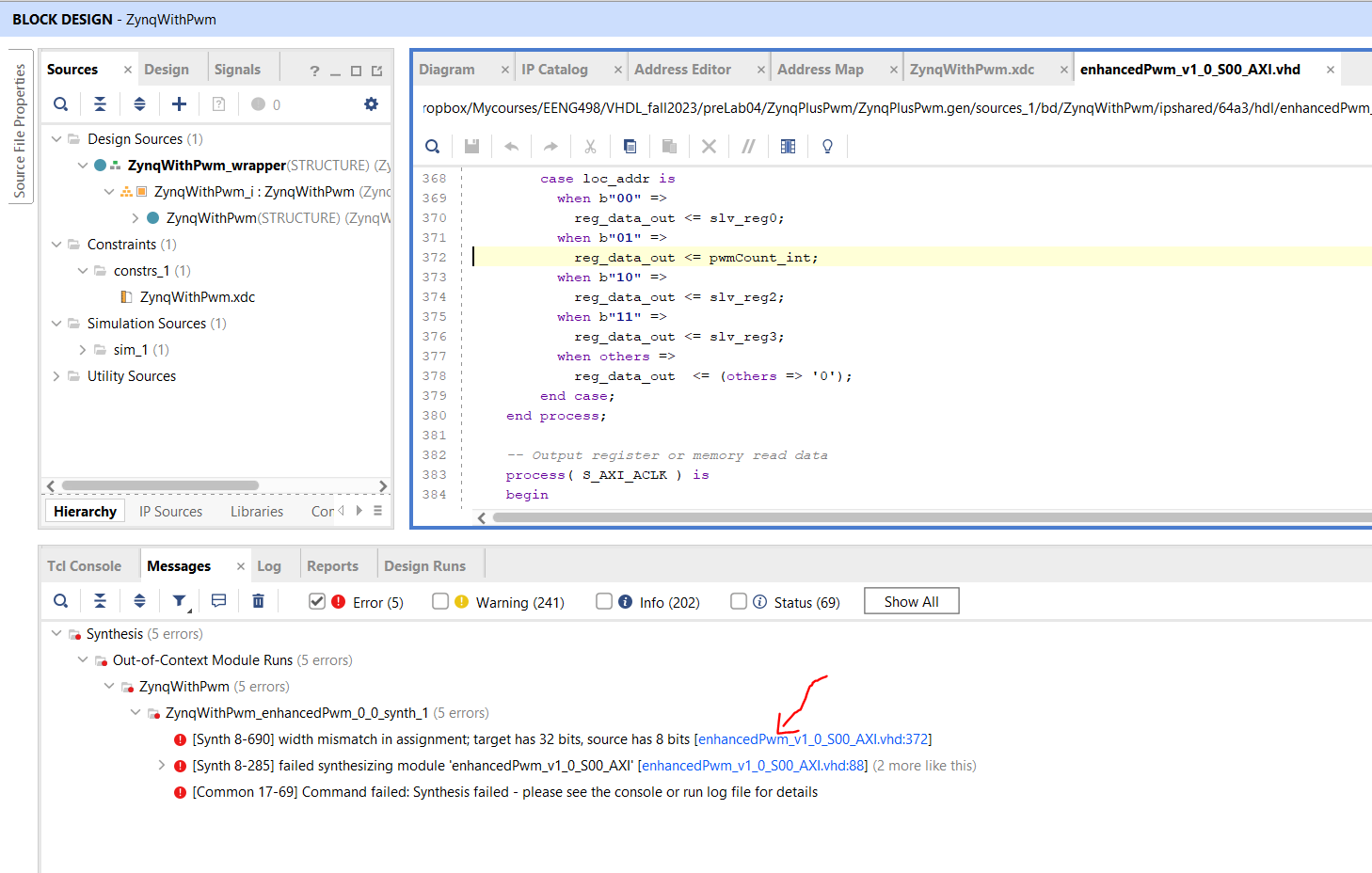


Open the zynqWithPwm.xdc file and type in the pin locations of the signals using the names as they appear in the block diagram. Most often these are the names that you used in the IP packager followed by “\_0”. Let’s route **enb\_ext\_0** to the push button PL\_KEY4. This push button is nominally logic 1 and equals logic 0 when pressed. This means that the enhancedPwm component will operate normally until the PL\_KEY4 button is pressed. Let’s route the **pwmSignal\_ext** to pin 3 on header J11. This will allow us to easily probe the output signal from the pwm and allow us to connect this output to some external circuitry (a low pass filter).

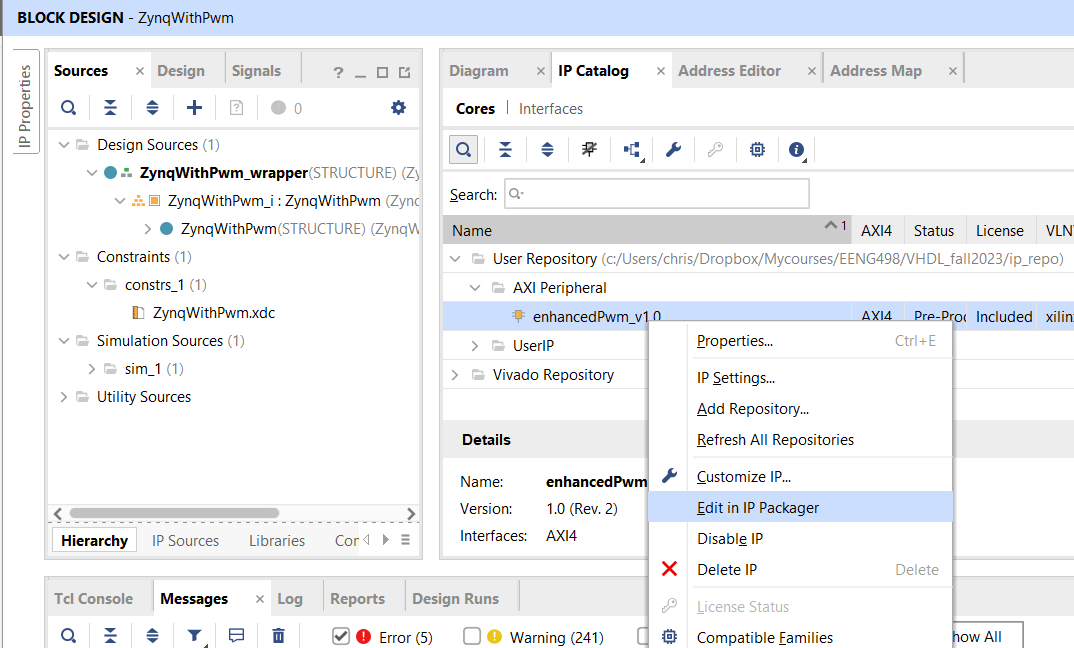


The last step in the process requires that you generate a bitstream by clicking Generate Bitstream in the Flow Navigator pane. If for some improbable reason you have not made a mistake somewhere you done, and take the time you just saved to go out and buy a lottery ticket. For the rest of us, sad face, keep reading.

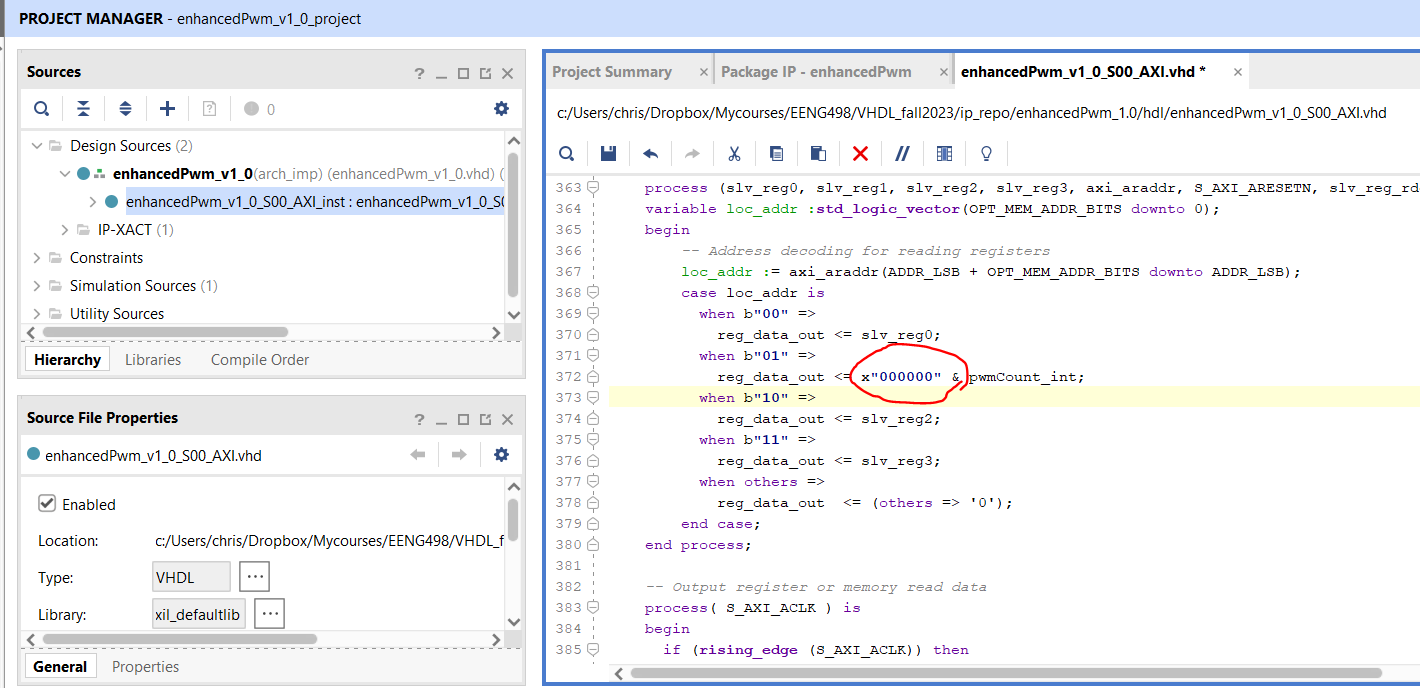
Invariably you will make errors in your IP that will cause the build to fail. For example, we forgot to make the 8-bit pwmCount\_int variable 32-bits so that it matches the 32-bit reg\_data\_out (see the error below). Let’s fix this now.



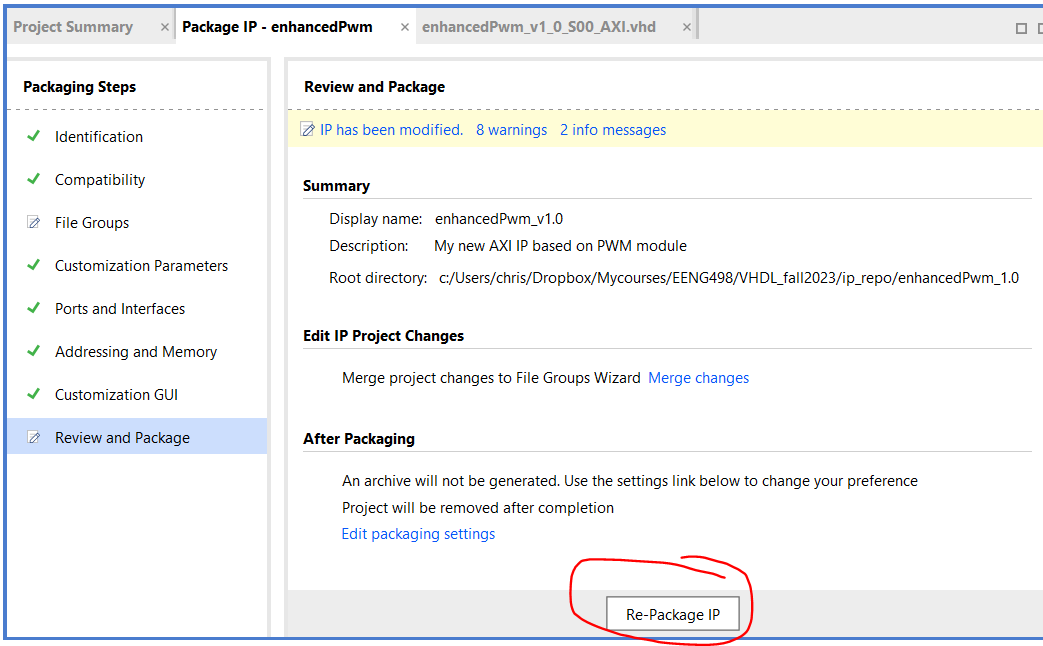
Open the IP Catalog (in left menu under Flow Navigator). Then open the User Repository, AXI Peripheral and then right click on the enhancedPwm and select Edit in IP Packager.



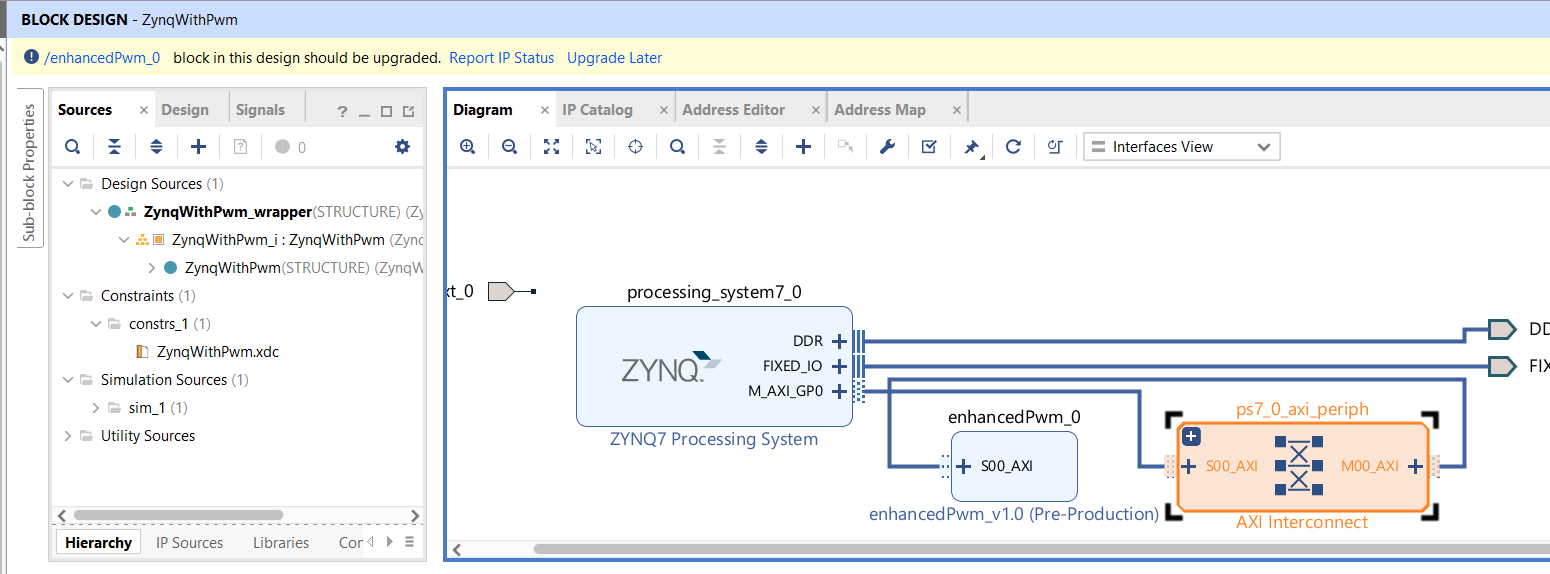
In the Edit in IP Packager pop-up, leave the defaults alone and click Ok. You will get a second Vivado invocation. Open the enhancedPwm\_v1\_0\_S00\_AXI.vhd file, find the troublesome assignment and concatenate a 24-bit zero vector in front of the pwmCounter\_int signal as shown below.



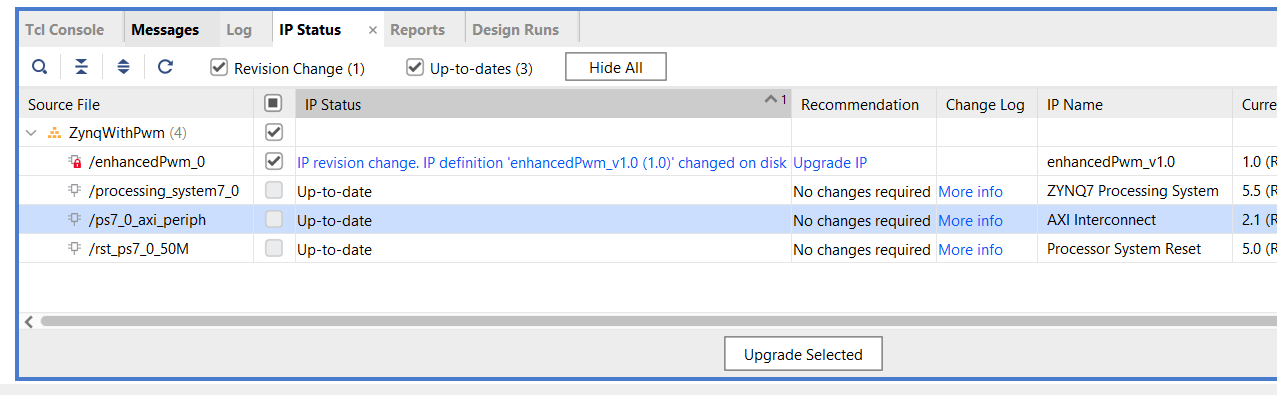
Save the file and then repackage your enhancedPwm IP using the last option in the Package IP tab.



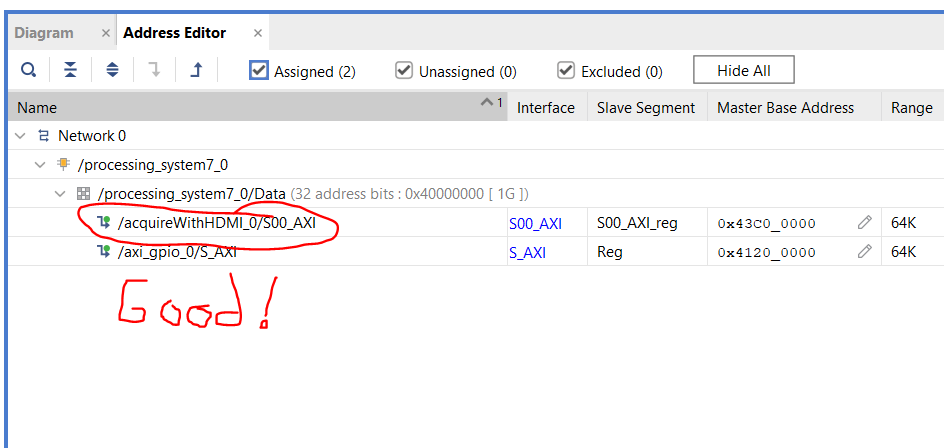
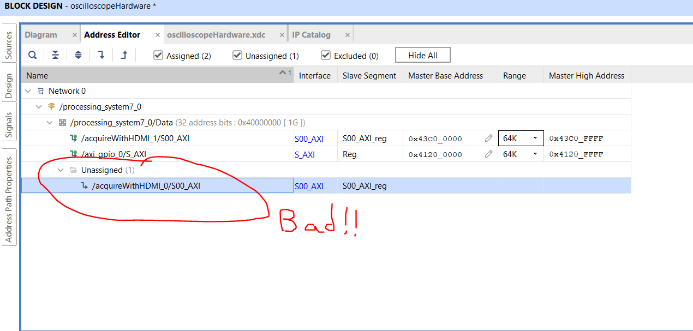
The second Vivado invocation will close when you complete the repackage. Back in your block diagram you will need to update the enhancedPwm IP. Do this as follows.



Start by clicking on the Report IP Status link. At the bottom of the page you should get a IP Status tab. Since the enhancedPwm\_0 module is selected, just click on the Upgrade Selected button.

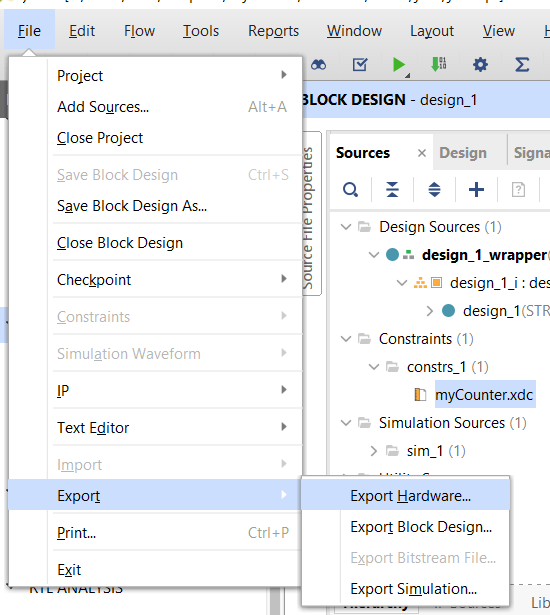


Click the Generate button on the Generate Output Product pop-up. When its complete, you can rerun the IP Status process to verify that everything is up-to-date. Before proceeding, please open the address tab and verify that Vivado has not unassigned your, modified, IP. You want the picture on the left. If you have the situation on the right, this is bad, right click on the custom IP and select assign.

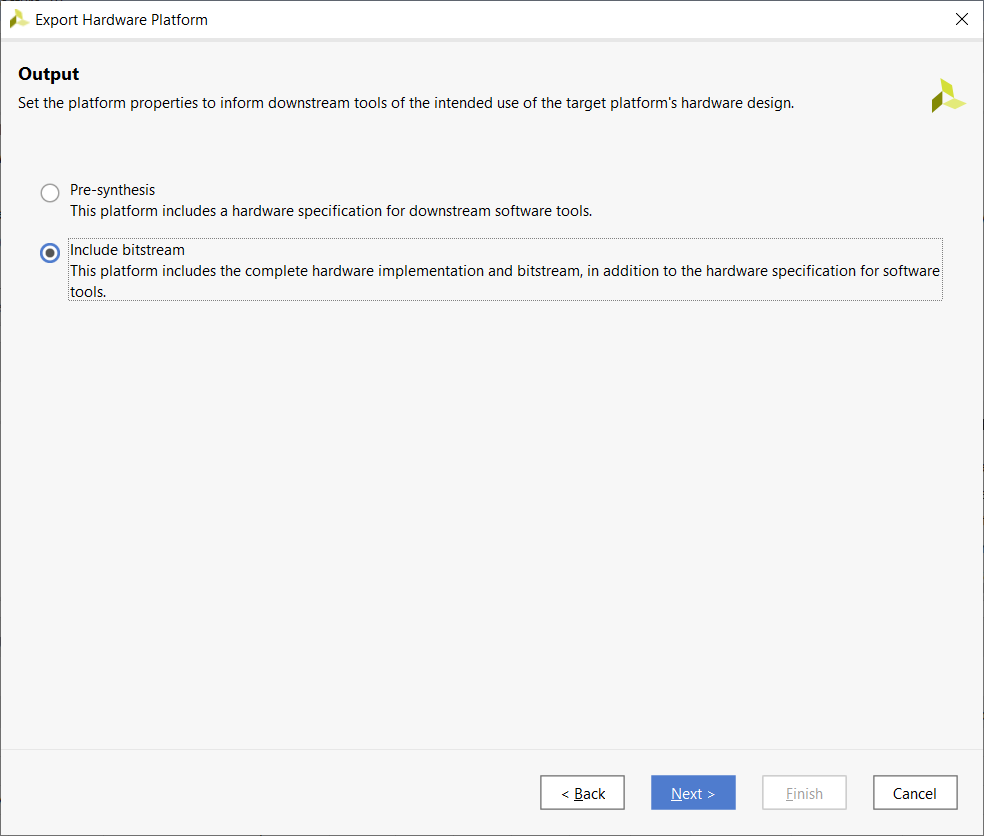


Now generate the Bitstream, find and correct errors, until everything is clean.

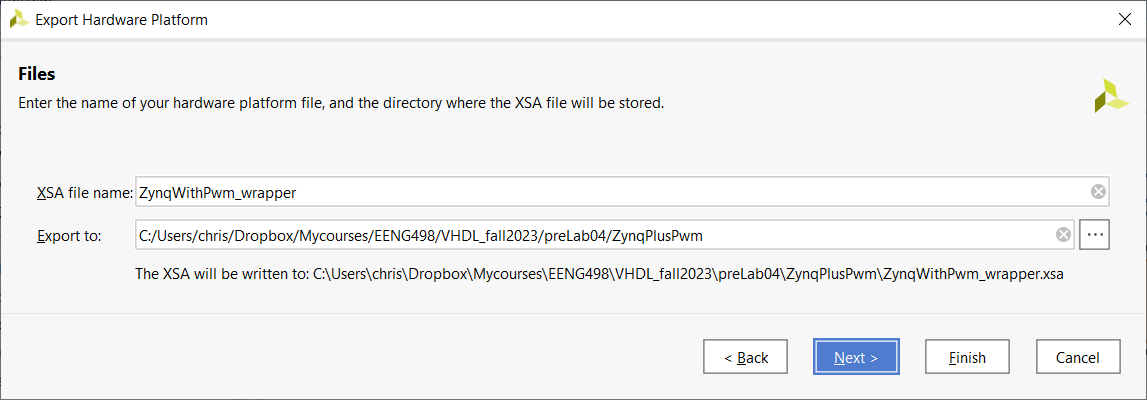
Once you get the design to successfully generate a bitstream, you will export the hardware from the File menu.



In the Export Hardware Platform Wizard, click Next on the first pop-up. Select the Include bitstream radio button in the Output step. Click Next.



Leave the defaults on the File step.



If you have already generated this design previously, overwrite the previous version. Click Finish and you are done with this phase. Next step will be to program the Zynq processor to interface with and utilize your custom IP.