# Lab 02: Creating AXI peripherals for your design

# Objectives

After completing this lab, you will be able to:

- Use the IP Packager feature of Vivado to create a custom peripheral for your IP

- Create a testbench to verify the functionality of your packaged IP

- Simulate the design using the Vivado simulator.

# Steps

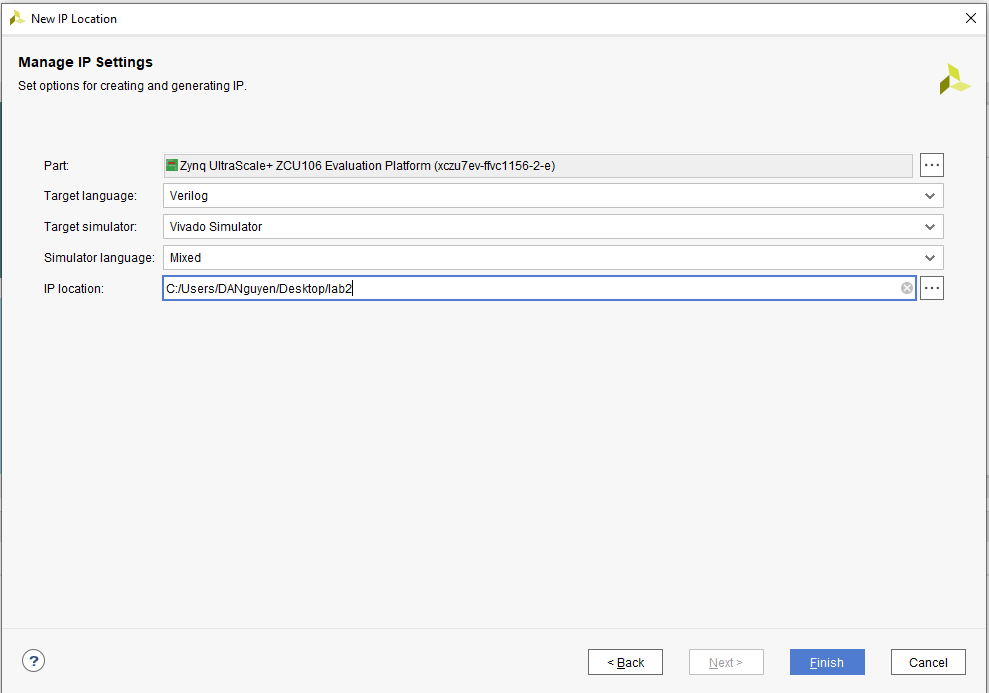
**Create a Custom IP using the Create and Package IP Wizard**

Launch Vivado and create an empty project targeting Zynq UltraScale+ ZCU106 Evaluation Platform, selecting Verilog as a target language. Use the provided lab1.v, and lab1\_zcu106.xdc (for ZCU106) files.

1. Open Vivado by selecting **Start > Xilinx Design Tools > Vivado 2021.2**.

2. Click **Manage IP** and select **New IP Location** and click *Next* in the *New IP Location* window

3. Select Verilog as the Target Language, Mixed as the Simulator Language, select the Part for ZCU106, and for IP location, type {labs}\lab2 and click Finish (leave other settings as defaults and click OK if prompted to create the directory)



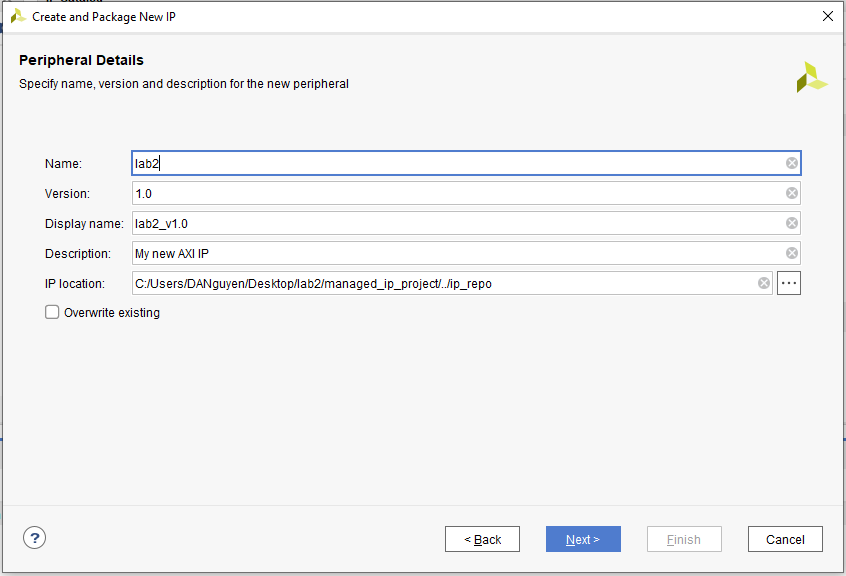
**Run the Create and Package IP Wizard.**

1. Select **Tools > Create and Package IP**
2. In the window, click **Next**
3. Select **Create a new AXI4 peripheral,** and click Next
4. Fill in the details for the IP

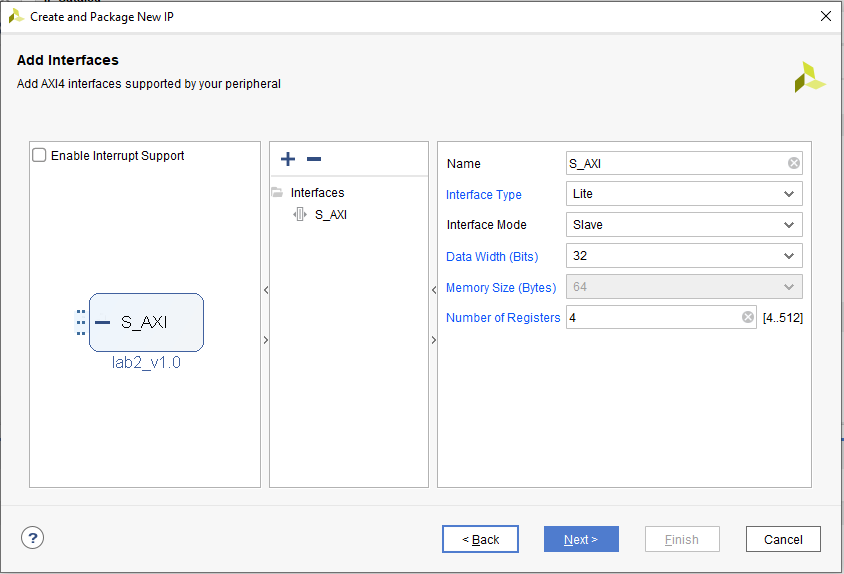
Name: lab2

Display name: lab2\_v1.0

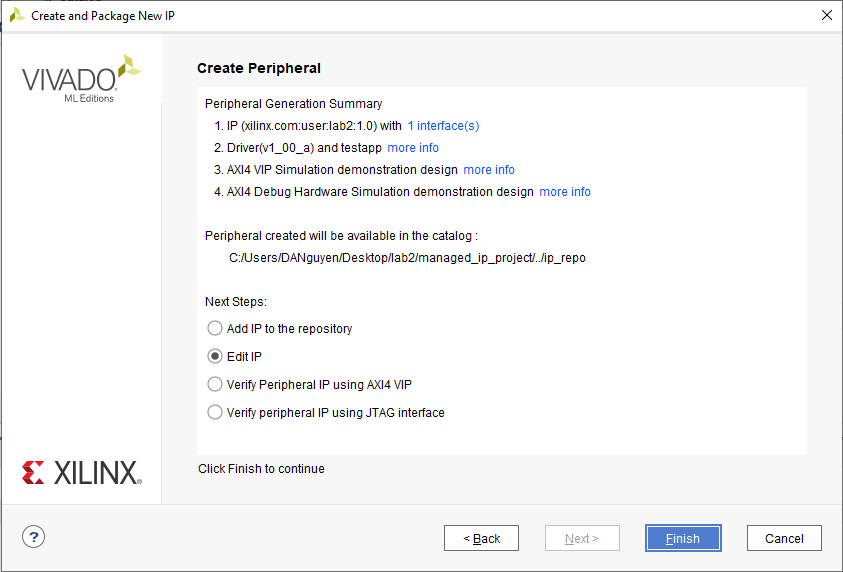
(Fill in description, Vendor Name and url)



1. Click Next
2. Change the Name of the interface to **S\_AXI**
3. Leave the other settings as default and click **Next** (Lite interface, Slave Mode, Data Width: 32, Number of Registers: 4)

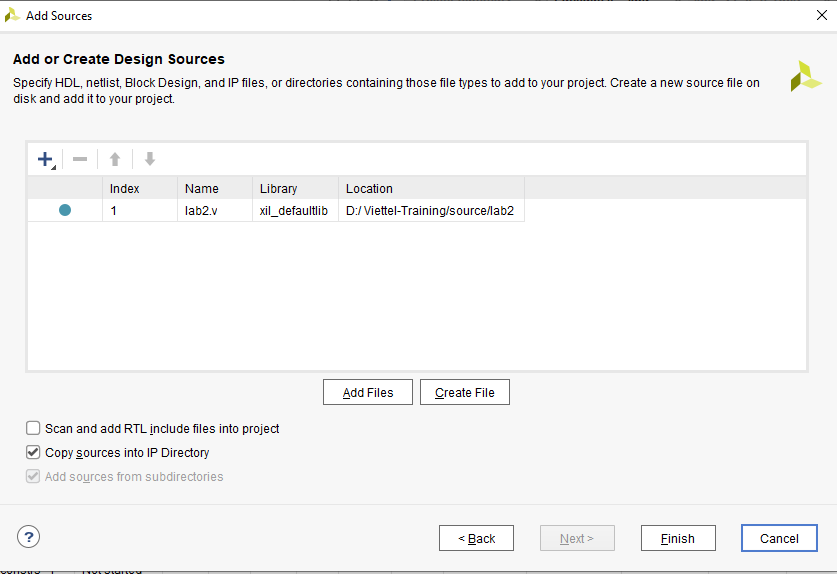


1. Select Edit Ip and click Finish (a new Vivado Project will open)

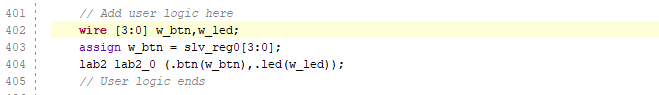


**Create an interface to the design.**

1. In the *Sources* pane, double click the *lab2\_ip\_v1\_0.v* file. This file contains the HDL code for the interface created above. The top-level file contains a module which implements the AXI interfacing logic, and an example design to write to and read from the number of registers specified above. This template can be used as a basis for creating custom IP.
2. In the Project Manager tab, click on *Add Sources* to add the design created in lab 1. The file is named lab2.v for convenience in this lab.

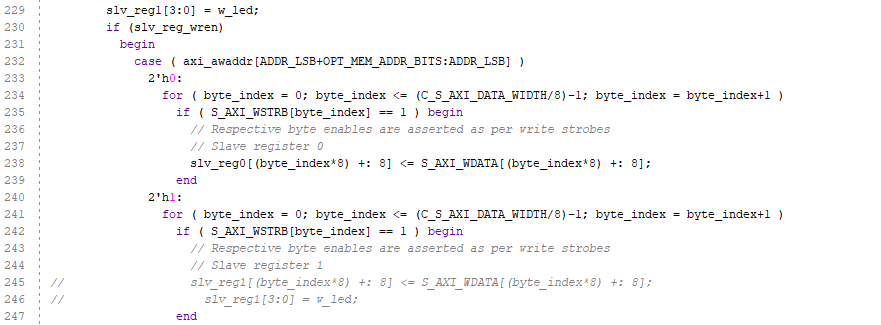


1. Open the lab2\_v1\_0\_S\_AXI.v files. Add the following lines at the end of the file



We have instantiated the design module in the AXI interface and connect the input to the design to the last 4 bits of the **slave register 0**

1. Add the following to connect the output of the design to the last 4 bits of the **slave register 1**



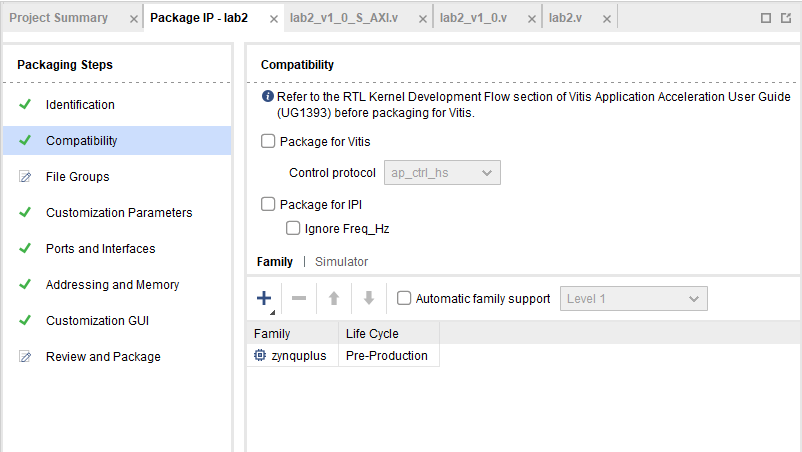
1. Save the file by selecting **File>Save File**
2. Click Run Synthesis and Save if prompted (This is to check the design synthesizes correctly before packaging the IP. If this was your own design, you would simulate it and verify functionality before proceeding).
3. Check the Messages tab for any errors and correct, if necessary, before moving to the next step. When Synthesis completes successfully, click Cancel.

**Package the IP.**

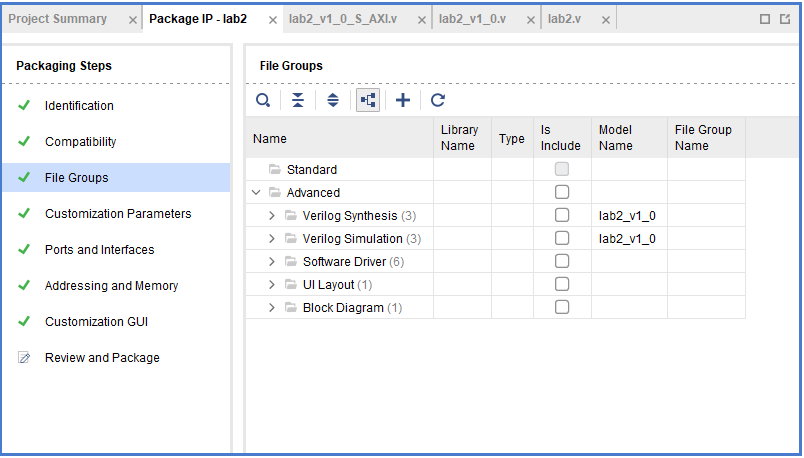
1. Click on the Package IP – lab2\_ip tab

Skip the next two steps (2 and 3) if you see /Basic\_Elements under the Categories section.

1. For the IP to appear in the IP catalog in particular categories, the IP must be configured to be part of those categories. To change which categories the IP will appear in the IP catalog click Blue Plus in the Categories section. This opens the Choose IP Categories window
2. For the purpose of the lab, uncheck the AXI Peripheral box and check the Basic Elements and click OK.
3. Select Compatibility. This shown the different Xilinx FPGA Families that the IP supports. The value is inherited from the device selected for the project.
4. Click the Blue Plus then Add Family Explicitly… from the menu
5. Select the Zynq Ultrascale +. You will get something like this:



1. You can also customize the address space and add memory address space using the IP Addressing and Memory category. We won’t make any changes.
2. Click on File Groups and click Merge changes from File Groups Wizard

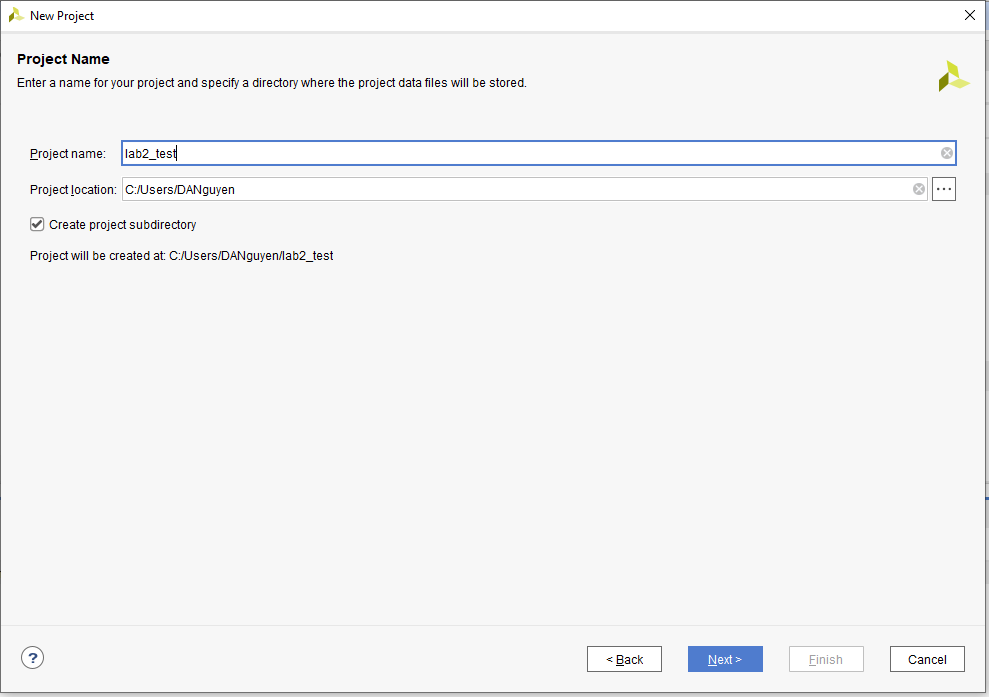


1. Select Review and Package, and notice the path where the IP will be created.
2. Click Package IP. Click Yes and the project will close when complete.
3. In the original Vivado window click File > Close Project

**Create Testbench**

1. Create a new project by File > New Project > Next.

2. Change the project name and click Next



3. Choose RTL project and click Next

4. In the next screen, change “target language” to “Verilog” and click Next.

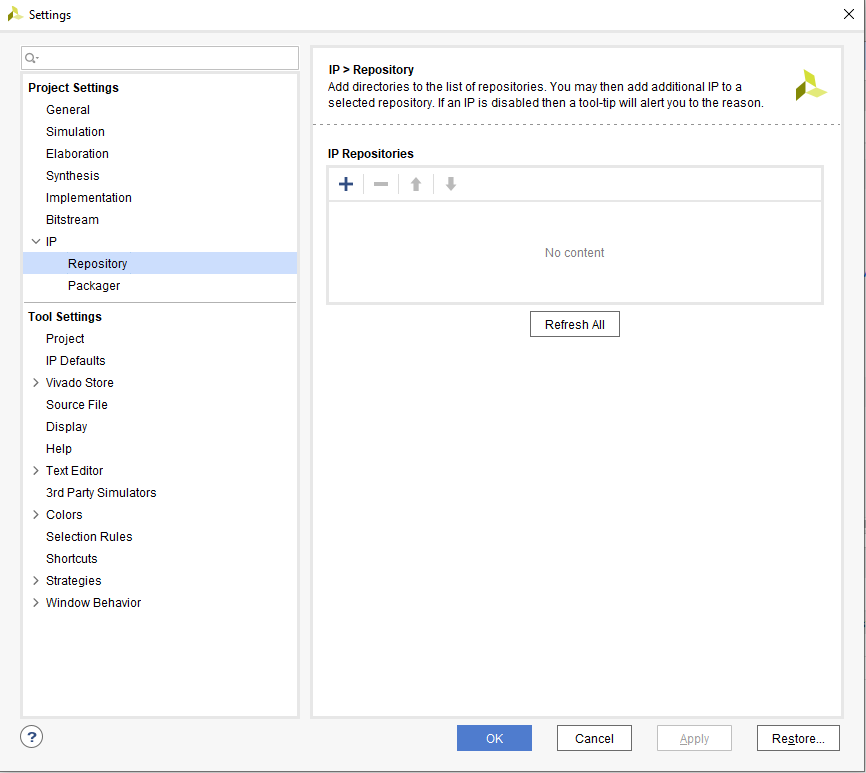
5. On “Add constraint screen”, click Next.

6. In the “default part screen”, click Board and type ZCU106 in the search. Then, select ZCU106 board and click next.

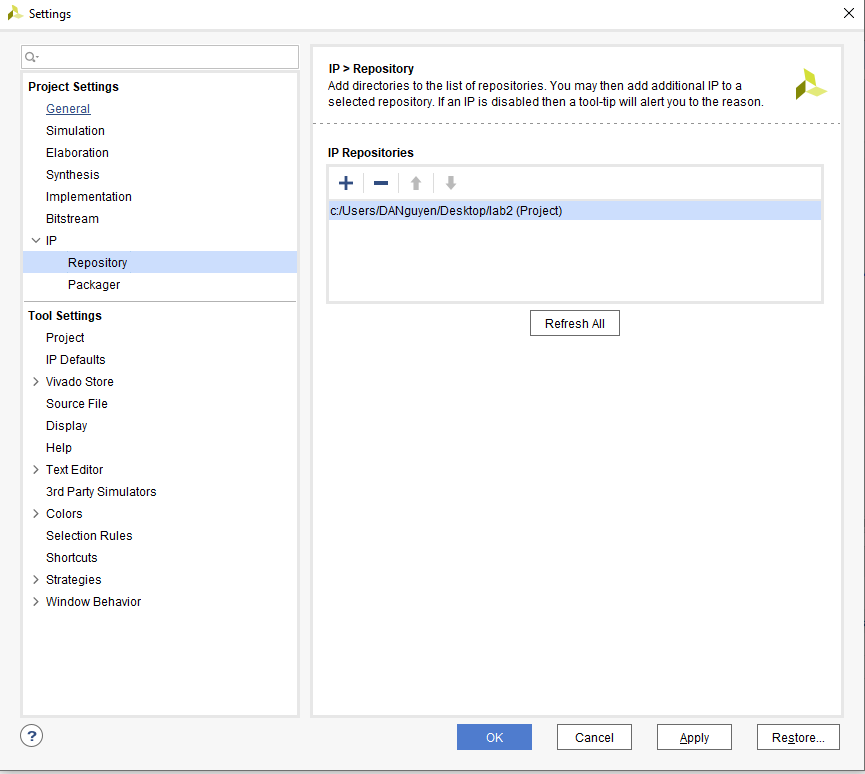
7. To finish creating the new project, click on “Finish”.

**Create the System Verilog testbench**

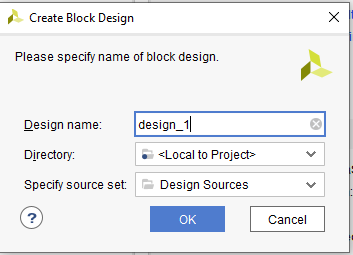
1. In “Flow Navigator”, click on “Settings” to add the IP to the project. Expand the IP catalog and select repository



2. Add the folder that we have created our IPs and click OK.

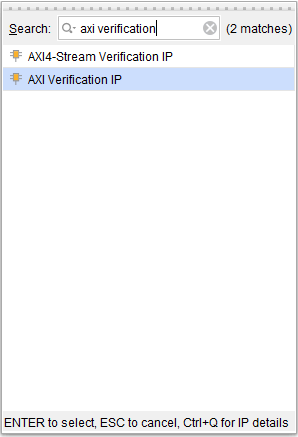


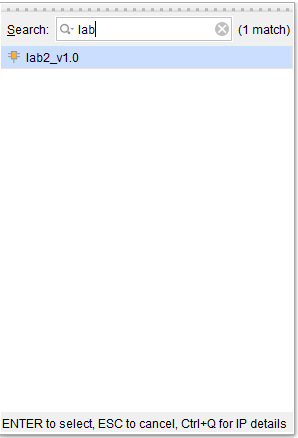
3. In “Flow navigator”, click on “Create block design” then select “OK”. A new board design is created.



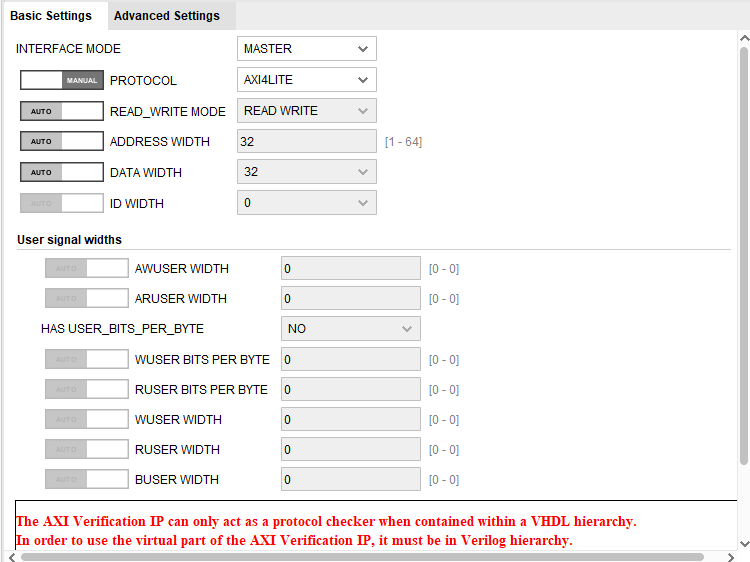
3. In “Flow navigator”, click on “Create block design” then select “OK”. A new block design is created.

4. Add “AXI Verification IP” and our lab2 IP by right click and choose “Add IP”.

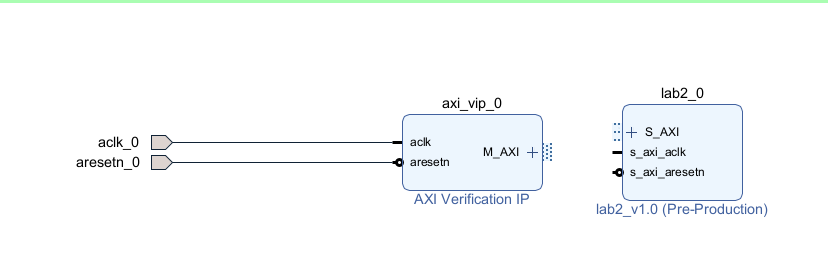




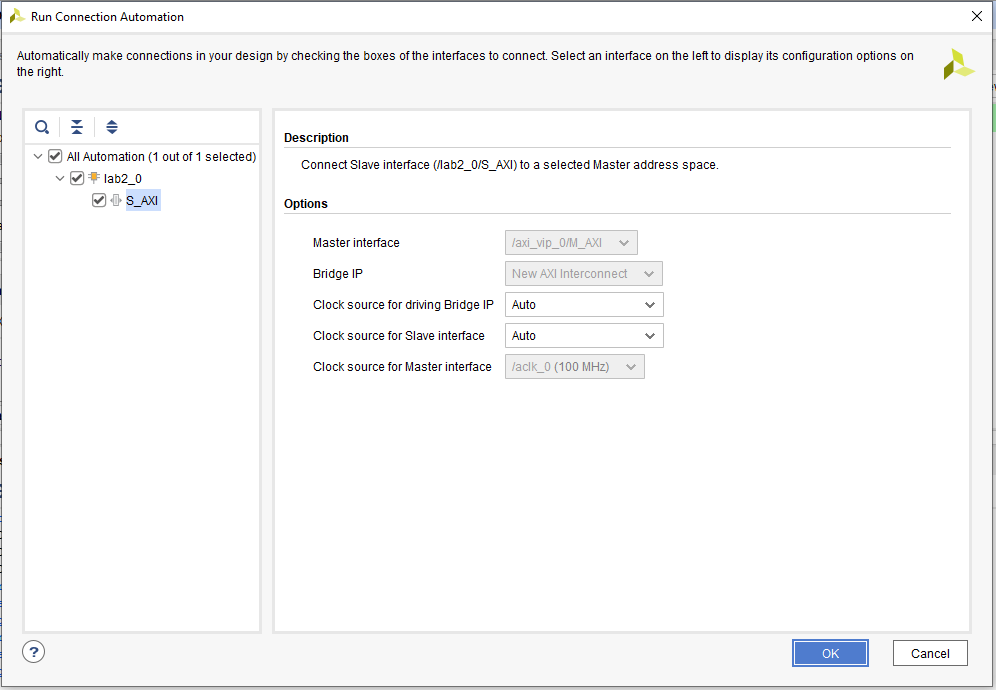
5. Customize “AXI Verification IP” by double click on AXI Verification IP and change the interface mode to MASTER, the protocol to AXI4LITE then click OK.

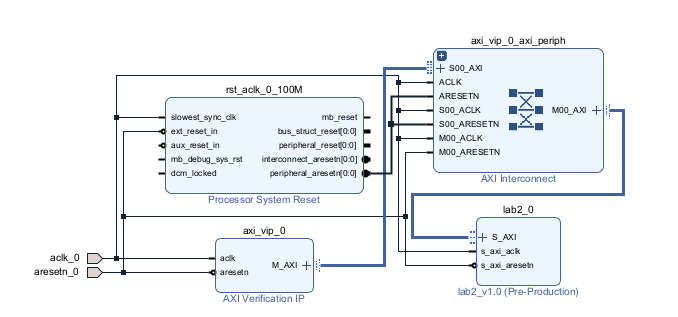


6. Right click on aclk of AXI Verification IP and choose “Make External”. Repeat this for the aresetn pin

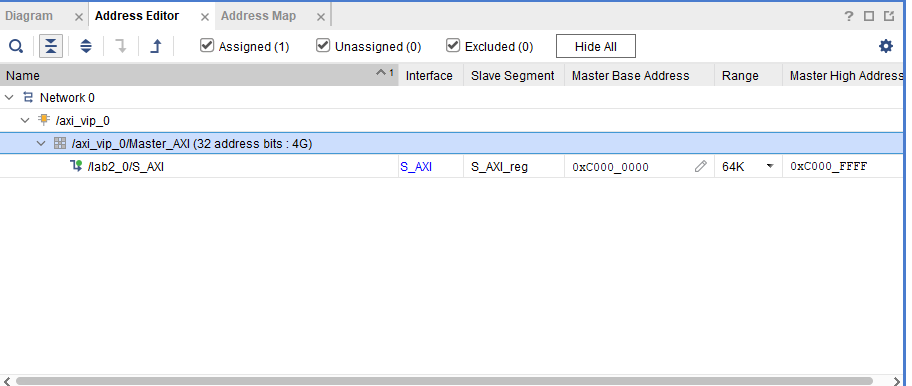


7. Connect the master AXI interface to the LED IP slave interface by using the Run connection Automation feature. Connect the aresetn pin to the ext\_reset\_in pin.

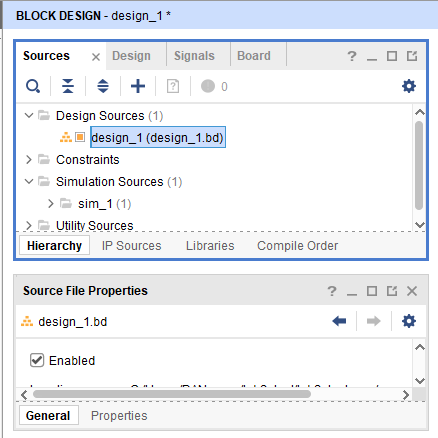




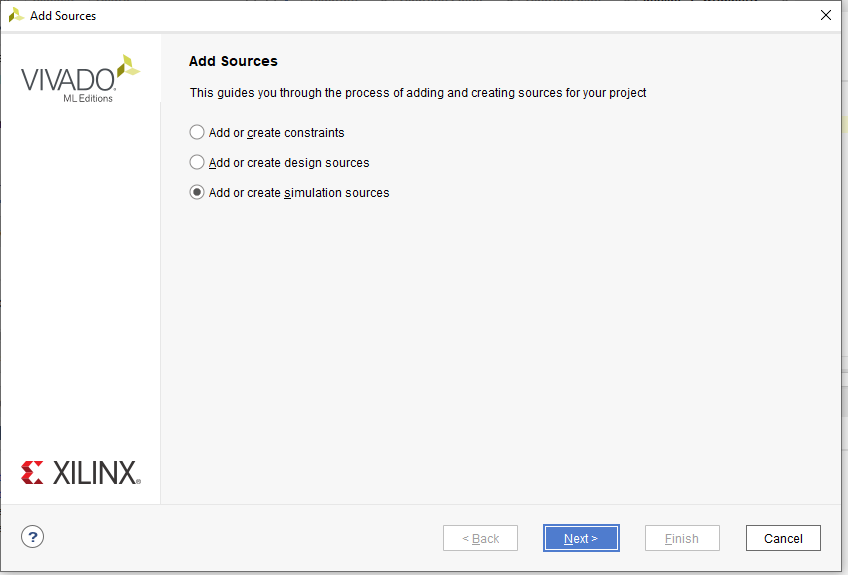
7. In the address editor, click on “Auto assign address button” and change the address of lab2 IP to “0xC0000000”



9. Right click on “design\_1” in the design sources, and choose “Create HDL wrapper” and click “OK”



10. Add a System Verilog source file by clicking on “Source” tab, right click on Sim\_1 and choose “add source file”



11. Choose “Add or create simulation sources” and click Next

12. Add the file “lab2\_ip\_tb.sv” in the source folder.

13. Run the simulation using XSIM as instructed in lab 1.

**Exercises.**

1. Study the testbench files. Make changes to the files so it can test all the combinations of the inputs and can be used as a self-checking testbench (Hint: You may refer to the sample testbench in lab 1).

2. Try to package the modified IP at the end of lab session 1. You may connect the new inputs and outputs to the register of your choice. Run the simulation with a self-checking testbench.

3. (Optional) Package the MAC IP in the optional part of lab 1. Use the following registers:

- Control Registers (Slave Register 0): Contains valid\_in signal at bit 0

- Status Registers (Slave Register 1): Contains valid\_out signal at bit 0

- Input Registers (Slave Register 2): Contains ai\_in at bit 0-7 and bi\_in at bit 8-15

- Output Registers (Slave Register 3): Contains mac\_out signal at bit 0-18

Write a self-checking testbench to test the MAC IP.