# Lab 04: Build a complete Embedded System with Vivado

# Objectives

After completing this lab, you will be able to:

- Create an embedded system desin using Vivado and Vitis

- Configure the Processing System (PS).

- Add Xilinx Standard IP in the Programmable Logic (PL) section

# Steps

**Create a Vivado Project**

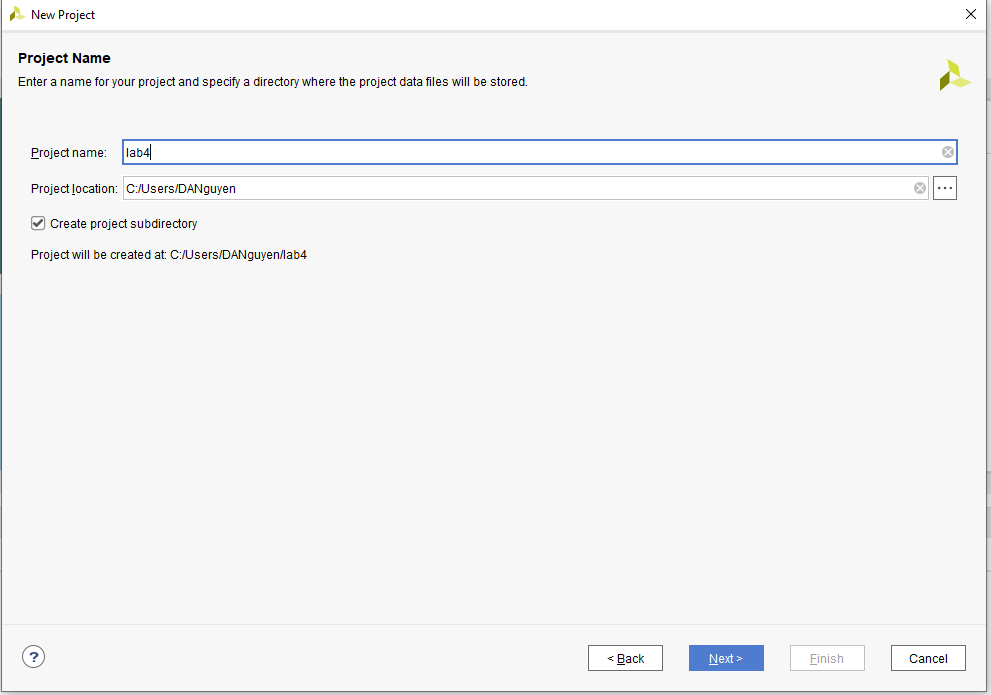
**Launch Vivado and create an empy project targeting the ZCU106 board, selecting Verilog as a target language**

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

2. Click **Create Project** to start the wizard. You will see the *Create A New Vivado Project* wizard page. Click **Next**.

3. Click the Browse button of the *Project Location* field of the **New Project** form, browse to the desired folder, and click **Select**.

4. Enter lab4 in the *Project Name* field. Make sure that the *Create Project Subdirectory* box is checked. Click **Next**.



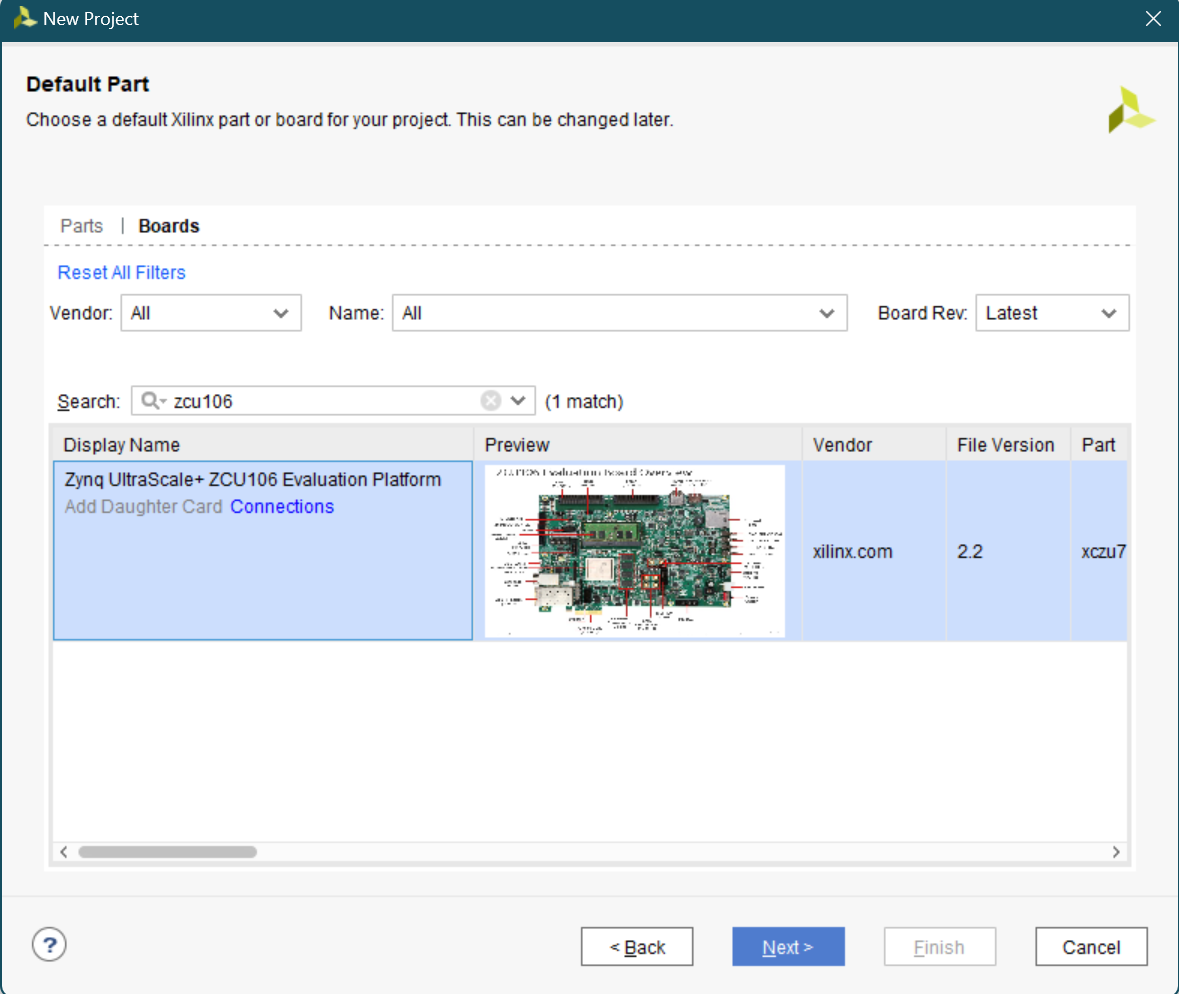
5. Select the **RTL Project** option in the Project Type form, and click **Next**.

6. Select **Verilog** as the *Target Language and Simulation Language* in the *Add Sources* form, and click **Next**.

7. Click **Next** to skip adding constraints.

8. In the *Default Part* form, click **Boards** filter.

9. In the *Boards* tab and then select **Zynq UltraScale+ ZCU106 Evaluation Platform**, and click **Next**.



10. Click **Finish** to create an empty Vivado project.

**Creating the Hardware System using IP Integrator**

1. Create a block design in the Vivado project using IP Integrator to generate the quad core Arm® Cortex™-A53 processing system (PS) and a dual core Arm Cortex-R5F real-time processor based hardware system.

2. In the Flow Navigator, click **Create Block Design** under IP Integrator.

3. Name the block system and click **OK**.

4. Click on the button.

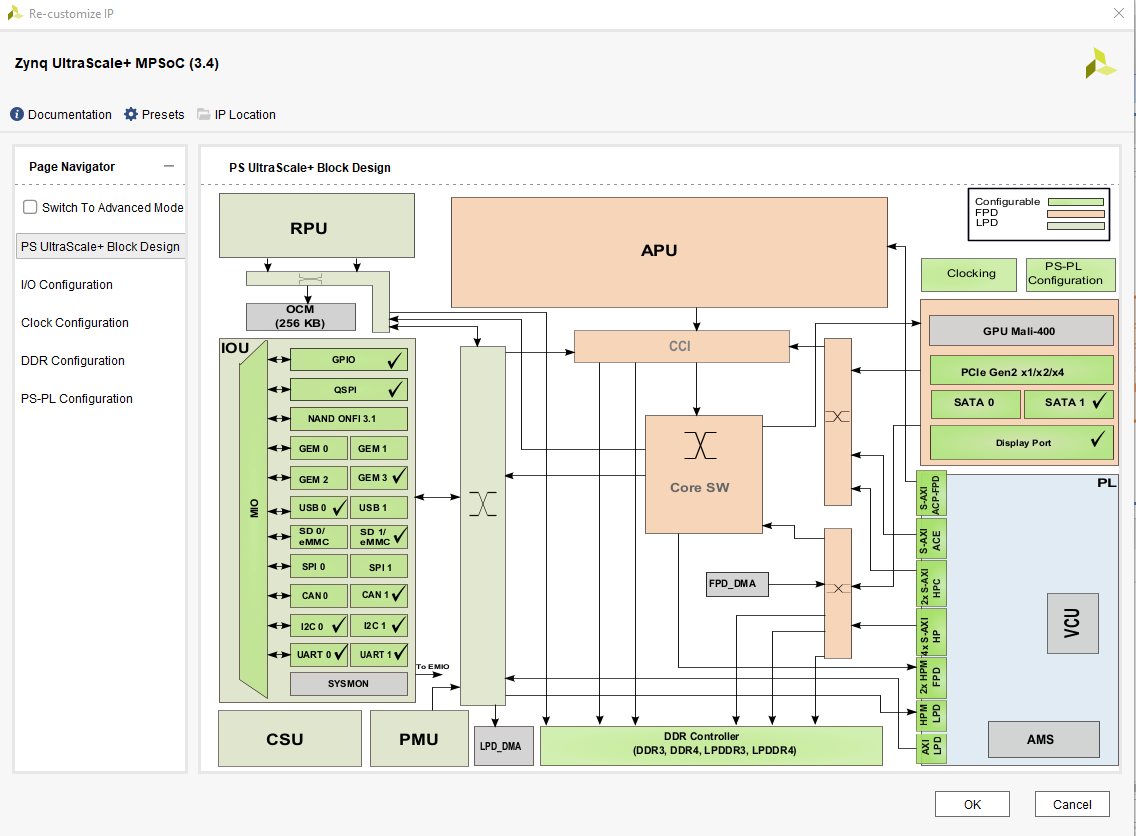
5. Once the IP Catalog is open, type zynq into the Search bar, and double click on the **Zynq UltraScale+ MPSoC** entry to add it to the design.

6. Click on **Run Block Automation** and click **OK** to automatically configure the board presets.

7. Double click on the Zynq block to open the *Customization* window for the Zynq UltraScale+ MPSoC.

A block diagram of the PS UltraScale+ should now be open, showing various configurable blocks of the Processing System.

8. At this stage, designer can click on various configurable blocks (highlighted in green) and change the system configuration.

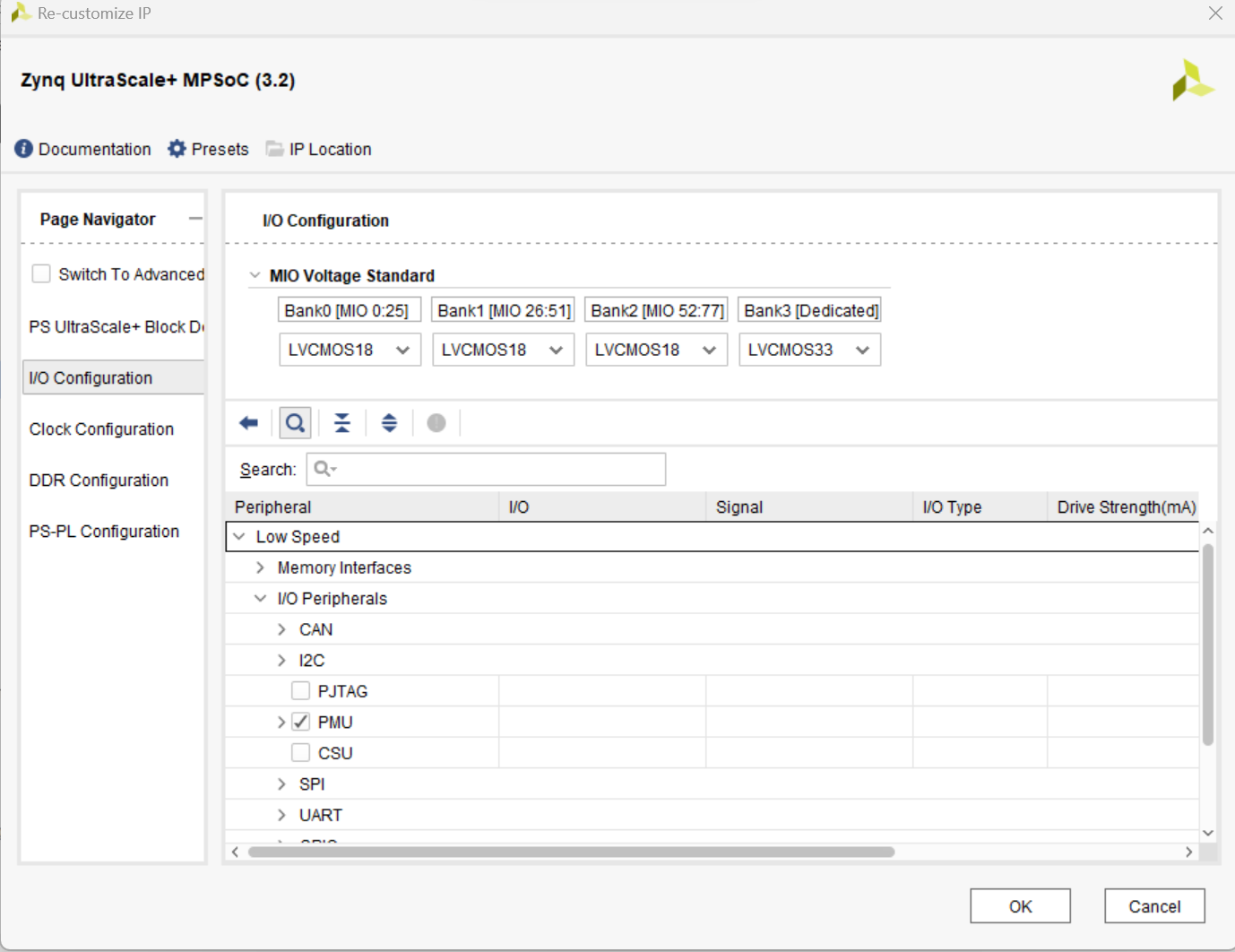


**Configure the I/O Peripherals block to have UART support.**

1. Click on the *I/O Configuration* panel to open its configuration form.

2. Expand the I/O Peripherals (and GPIO).

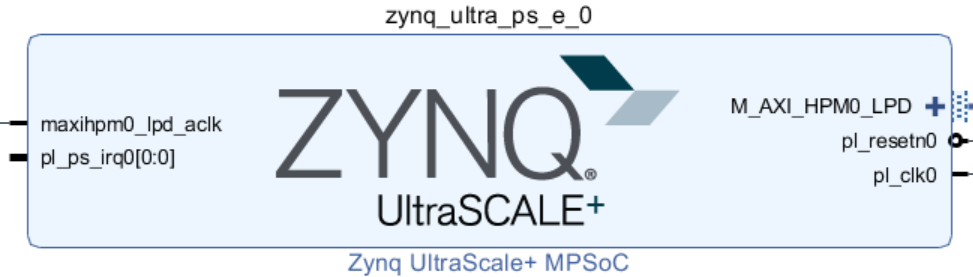
3. Deselect CAN and GPIO peripherals except *UART*.



4. Select *PS-PL Configuration > PS-PL interfaces > Master interface*, disable AXI HPM0 FPD and AXI HPM1 FPD, then enable the AXI HPM0 LPD option.

5. Click **OK**.

The configuration form will close and the block diagram will be updated as shown below.



**Add one instance of GPIO, name it buttons, and configure for the board. Connect the block to the Zynq.**

1. Click the button and search for **AXI GPIO** in the catalog.

2. Double-click the **AXI GPIO** to add an instance of the core to the design.

3. Click on the **AXI GPIO** block to select it, and in the *Block properties* tab, change the name to **buttons**.

4. Double click on the **AXI GPIO** block to open the customization window. Under *Board Interface*, for GPIO, click on **Custom** to view the drop-down menu options, and select **push buttons 5bits** for the ZCU106 board.

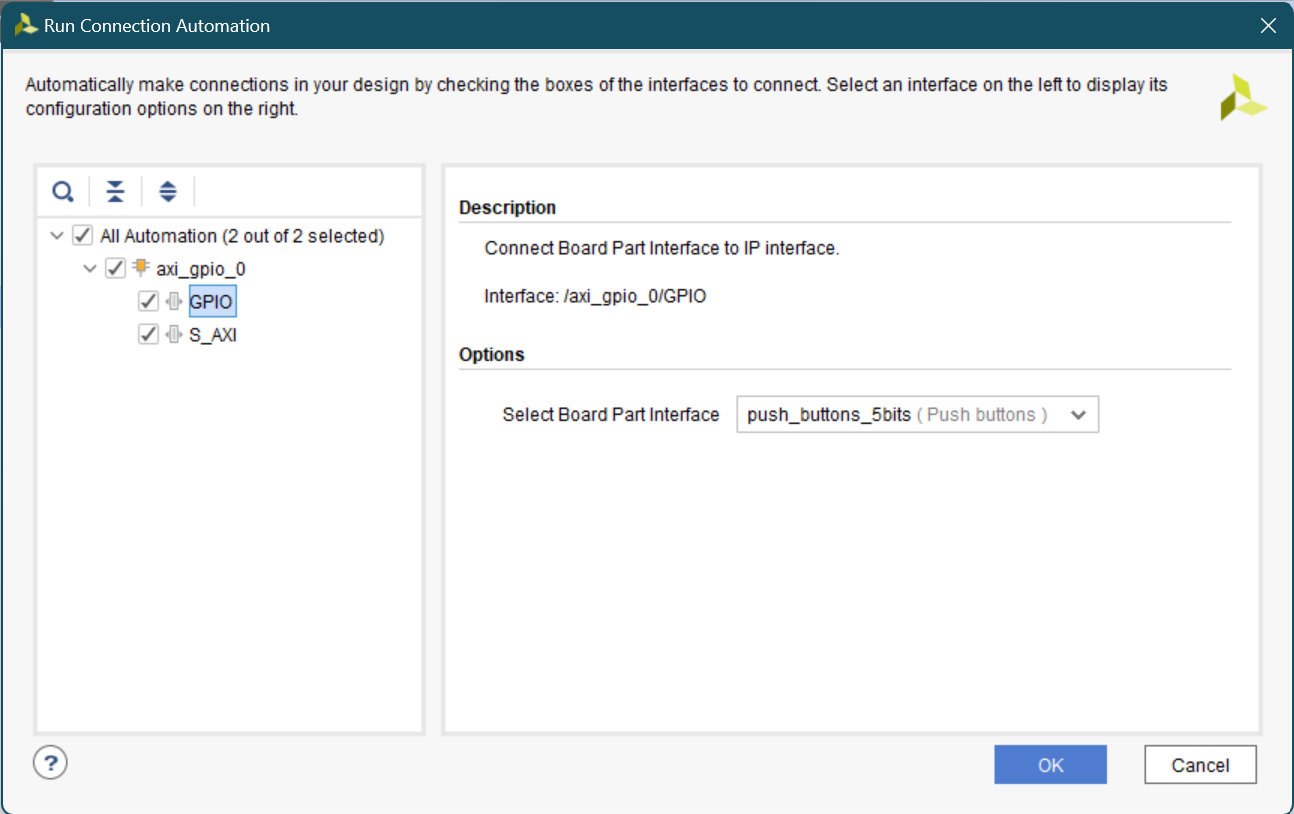
As the board was selected during the project creation, and a board support package is available for these boards, Vivado has knowledge of available resources on the board.

5. Click the **IP Configuration** tab. Notice the GPIO Width is set to 5 and is greyed out. If a board support package was not available, the width of the IP could be configured here.

6. Click **OK** to finish configuring the GPIO and to close the *Re-Customize IP* window.

7. Click on **Run Connection Automation**, and select **buttons** (which will include GPIO and S\_AXI)

Click on **GPIO** and **S\_AXI** to check the default connections for these interfaces.



8. Click **OK** to automatically connect the *S\_AXI* interface to the Zynq *GP0* port (through the AXI interconnect block), and the GPIO port to an external interface.

Notice that after block automation has been run, two additional blocks that are required to connect the blocks, *Processor System Reset*, and *AXI Interconnect* have automatically been added to the design.

9. Add another instance of GPIO, name the instance *leds*, configure it and connect it to the Zynq.

10. Add another instance of the *GPIO* peripheral.

11. Change the name of the block to **leds**.

12. Double click on the *leds* block, and select **leds 8bits** for the *GPIO* interface and click **OK**.

13. Click on **Run Connection Automation**

14. Click **leds**, and check the connections for *GPIO* and *S\_AXI* as before

15. Click **OK** to automatically connect the interfaces as before.

Notice that the AXI Interconnect block has the second master AXI (M01\_AXI) port added and connected to the S\_AXI of the leds.

16. Add another instance of the *GPIO* peripheral.

17. Change the name of the block to **switches**.

18. Double click on the *switches* block, and select **dip switches 8bits** for the *GPIO* interface and click **OK**.

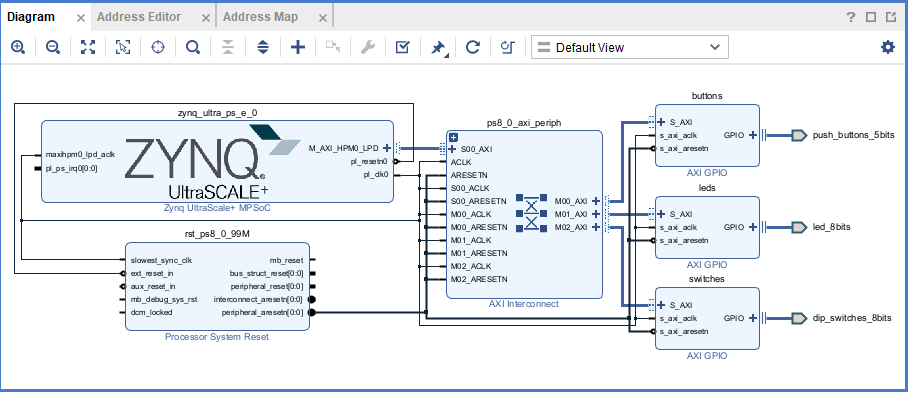
19. Click on **Run Connection Automation**

20. Click **switches**, and check the connections for *GPIO* and *S\_AXI* as before

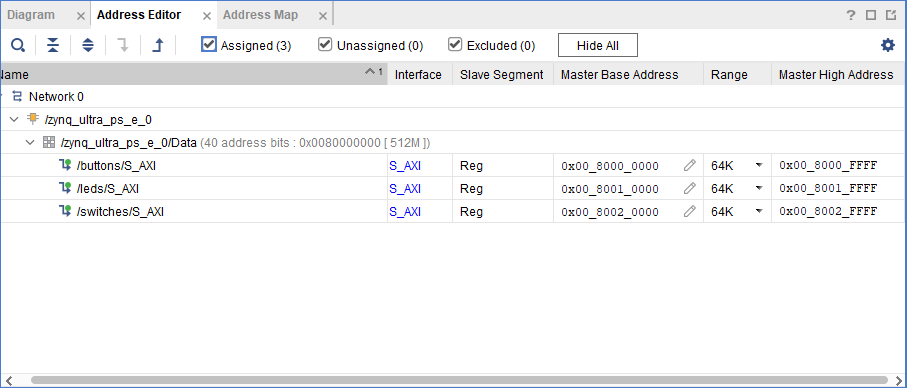
21. Click **OK** to automatically connect the interfaces as before.

Notice that the AXI Interconnect block has the third master AXI (M02\_AXI) port added and connected to the S\_AXI of the leds.

At this stage the design should look like as shown below.



22. Select the **Address Editor** tab and see that the addresses are assigned to the three GPIO instances. They should look like as follows.



23. Select the Diagram tab, and click on the alt tag (Validate Design) button to make sure that there are no errors.

Ignore warnings.

24. Select **File > Save Block Design** to save the design.

25. Since all IO pins are board-aware no additional user constraints are need.

**Generate the Bitstream.**

1. In Vivado, select the *Sources* tab, expand the *Design Sources*, right-click the *lab4.bd* and select **Create HDL Wrapper…**

2. Click OK when prompted to allow Vivado to automatically manage this file.

The wrapper file, lab4\_wrapper.v, is generated and added to the hierarchy. The wrapper file will be displayed in the Auxiliary pane.

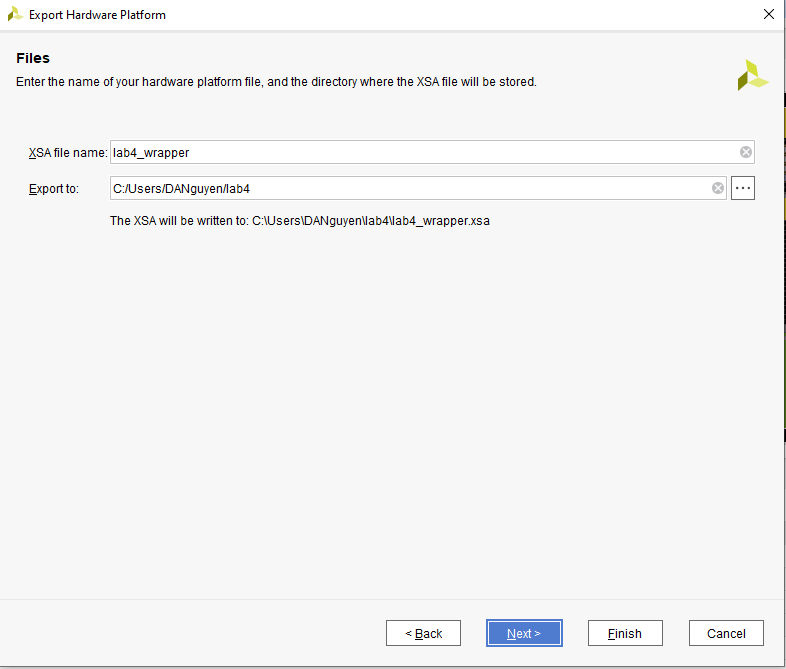
3. Click on the **Generate Bitstream** in the *Flow Navigator* pane to synthesize and implement the design, and generate the bitstream. Click **Save** and **Yes** if prompted. Click OK to launch the runs.

4. When the bitstream generation is complete, click **Cancel**.

**Export the Hardware.**

1. Export the hardware configuration by clicking **File > Export > Export Hardware…** Tick the box to include the bitstream and click **OK**.

2. Check the location where the XSA file will be saved.



3. Click **Finish**.

**Exercises**

1. Try to make a new hardware platform. This time include the IP you have made in lab 2. You may make some modifications to the IP, such that the output LEDs can be connected to the outside world and this will be connected to the LEDs of the board.

2. (Optional) Make a new hardware platform including the MAC IP you have developed in lab 2.