

Lab-1 (7th Aug & 9th Aug)

ECE270 - Embedded Logic Design

August 4, 2017

1 Tasks to be done in this lab including homework.

1. Design and implement a half-adder (using data flow approach) on the Zedboard.
2. Using the half-adders designed in step 1, implement a full-adder.
3. Implement a 4-bit adder/subtractor (adds/subtracts two 4-bit inputs) using the full-adders designed in step 2.

Note: All the the designs must be verified on the Zedboard. Before next lab all the designs must be completed.

2 Step-by-step guide for implementing half-adder.

All the steps will be discussed in detail during the lab. Follow the steps shown in figures one-by-one.

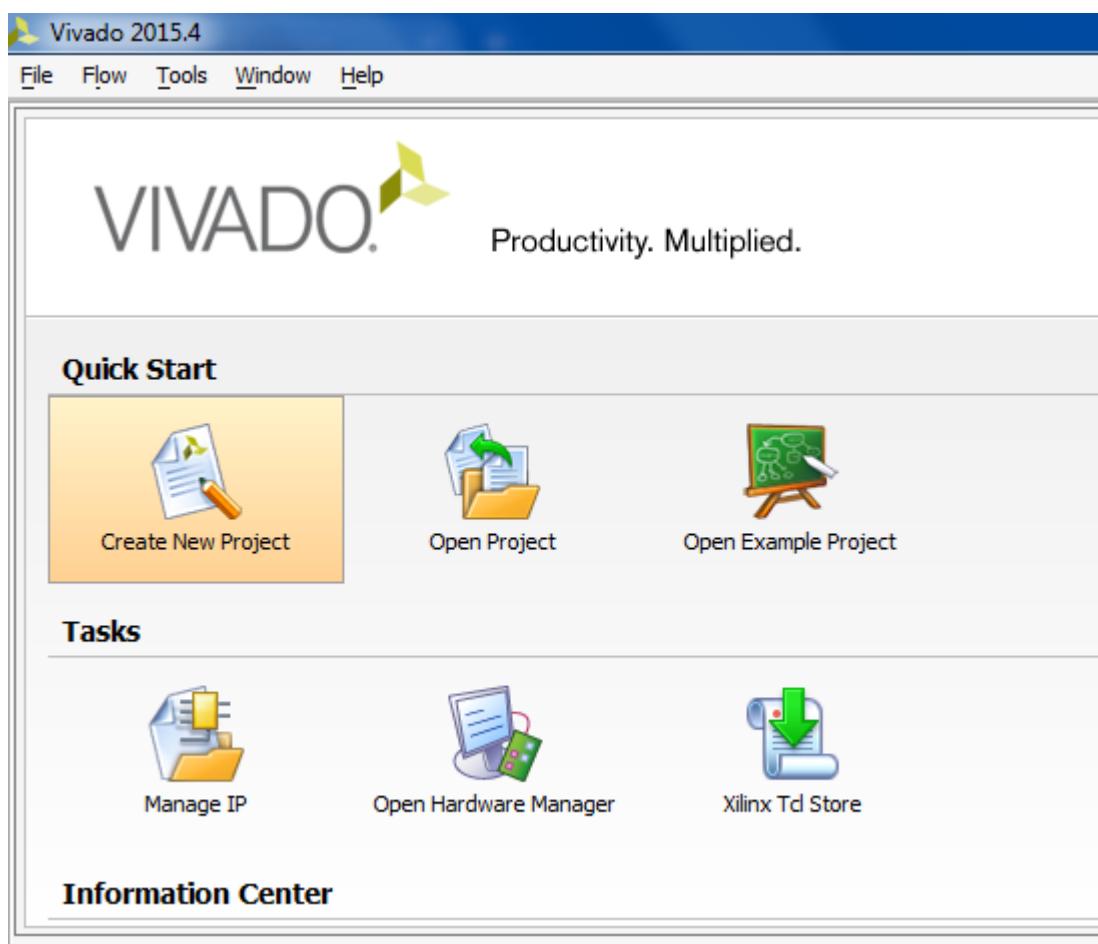


Figure 1: Create New Project

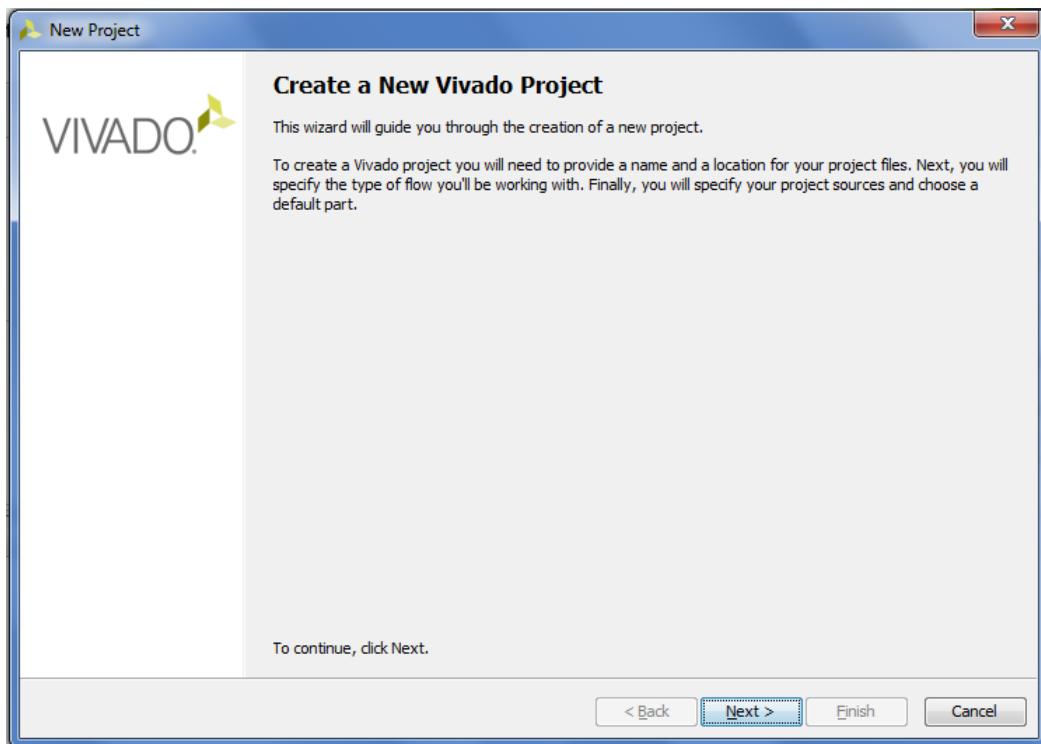


Figure 2: Click Next

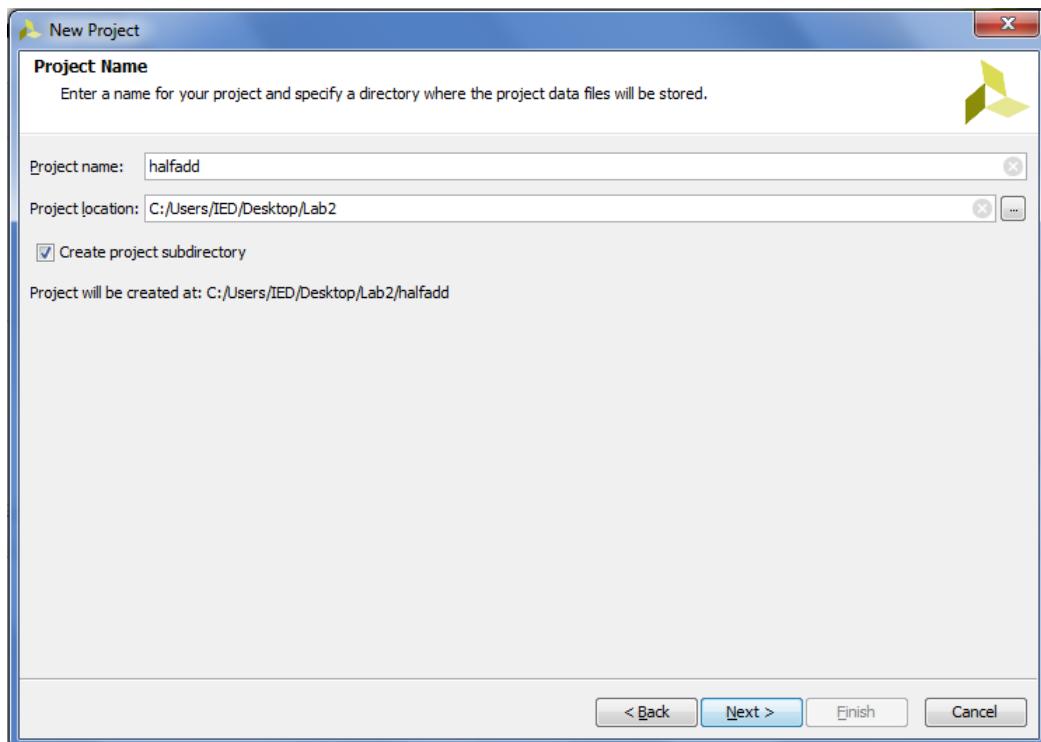


Figure 3: Give your project a name and set the location of your project

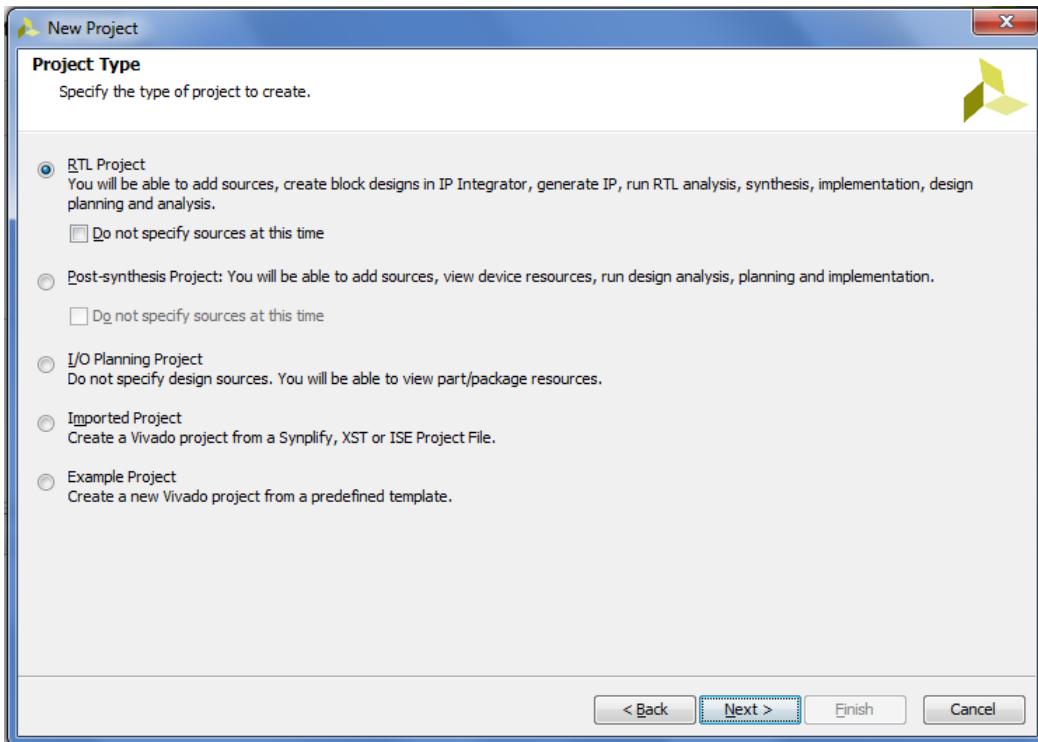


Figure 4: Select the type of project you wish to create (**RTL Project** in our case, uncheck the option “Do not specify...”)

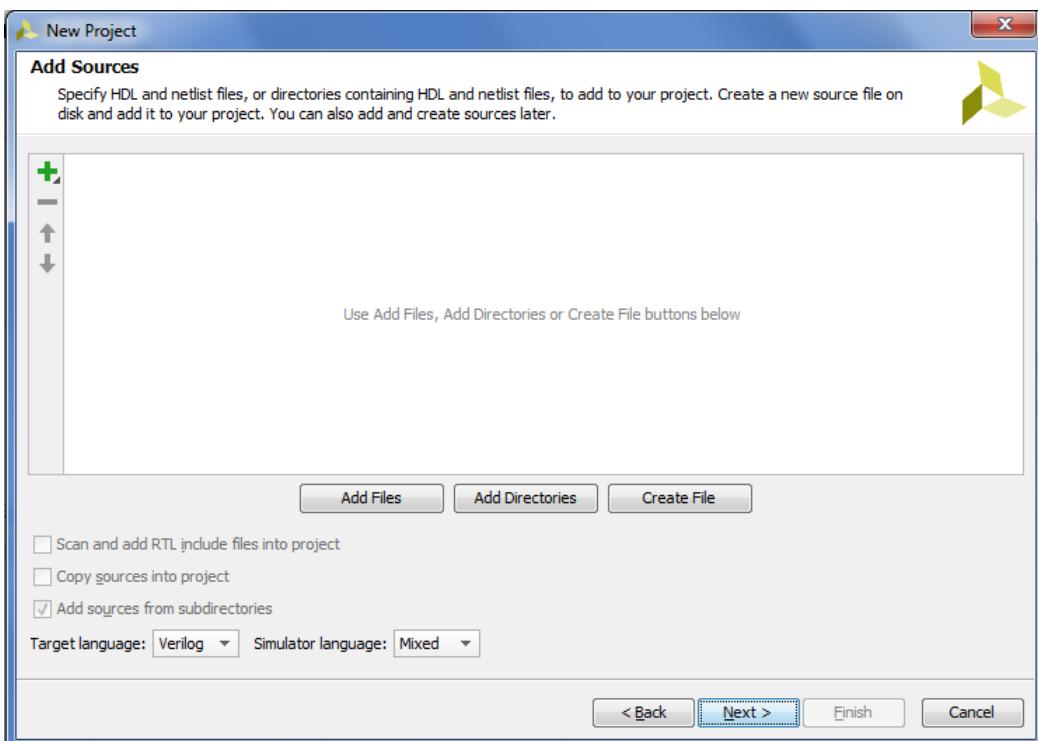


Figure 5: In Add Sources, click on **Create File** to create new files

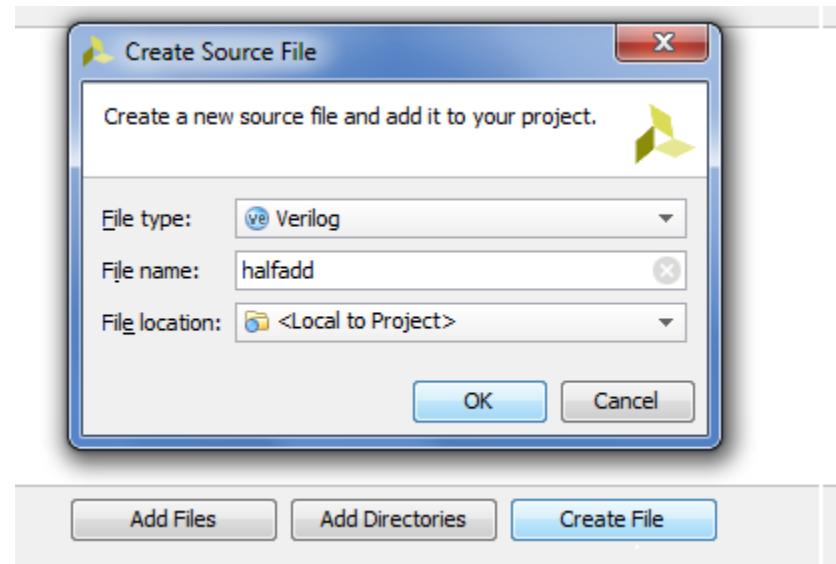


Figure 6: Give a suitable name to your verilog file

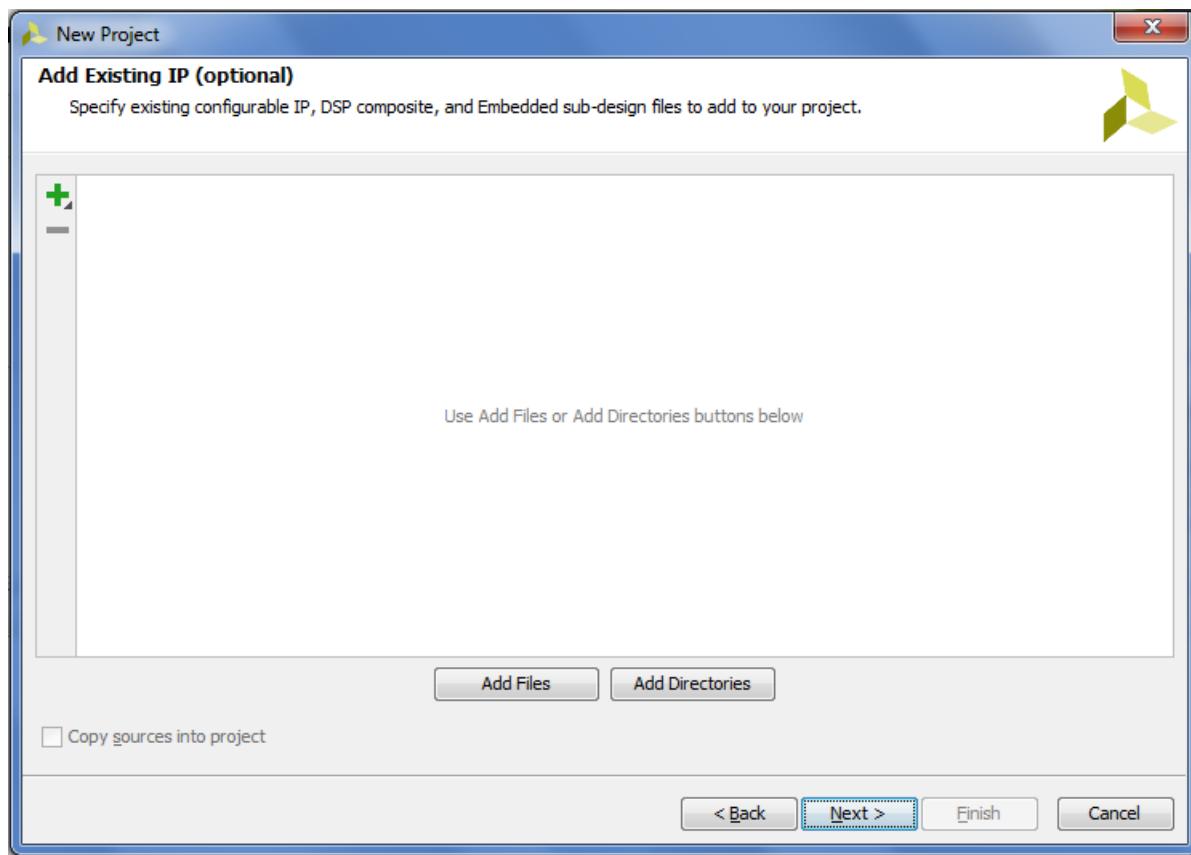


Figure 7: Add Existing IPs if required (click **Next >** for our case)

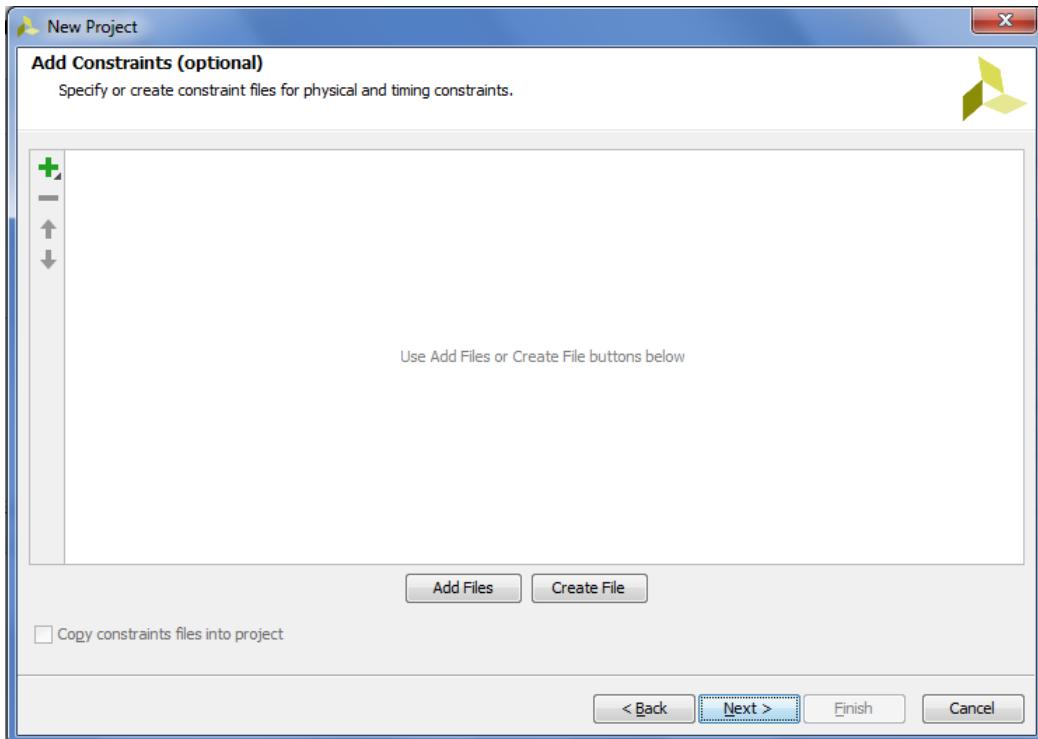


Figure 8: Add Constraints if required (click **Next** for our case)

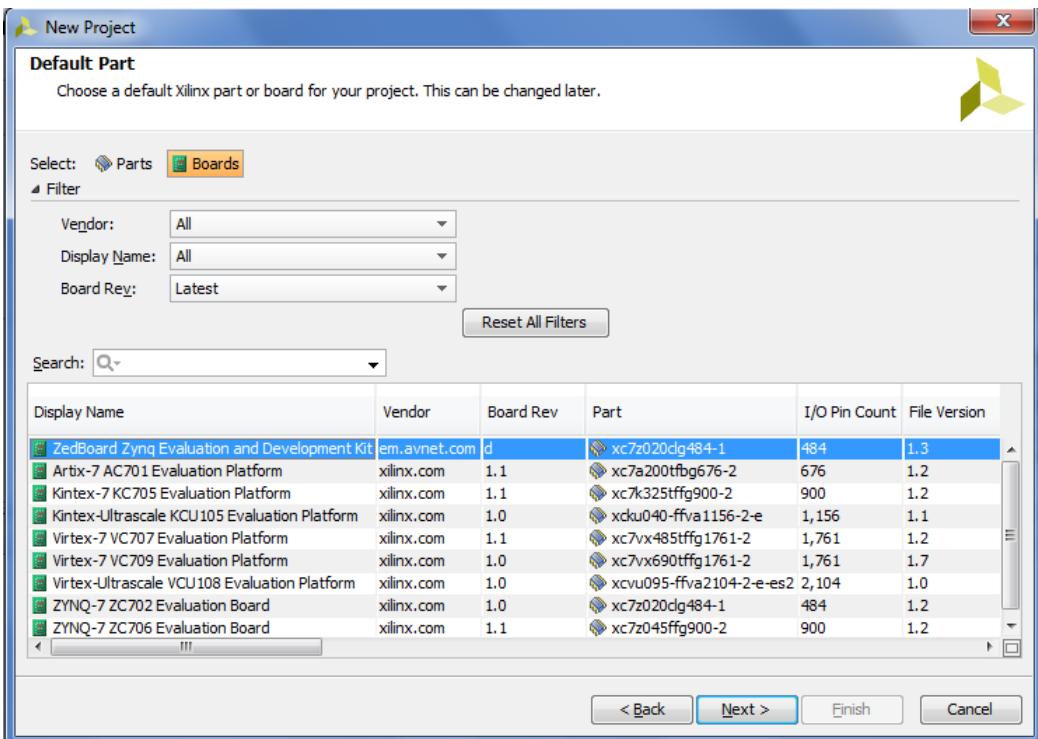


Figure 9: Select the target FPGA device (Zedboard in our case)

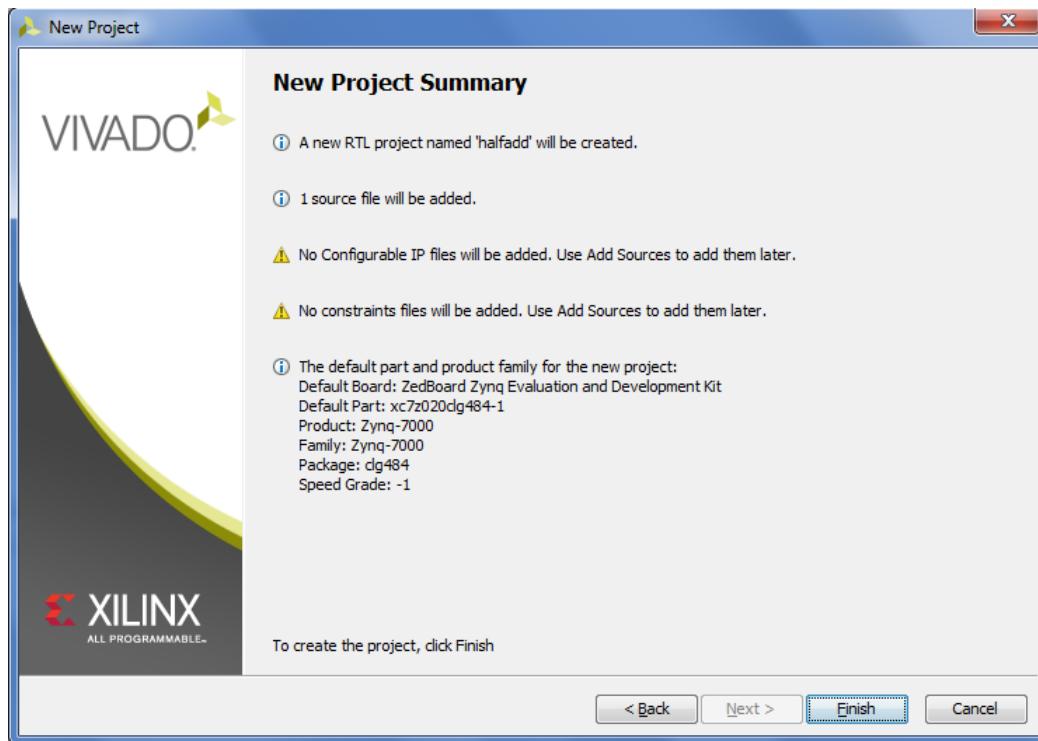


Figure 10: Check the project summary, click **Finish**.

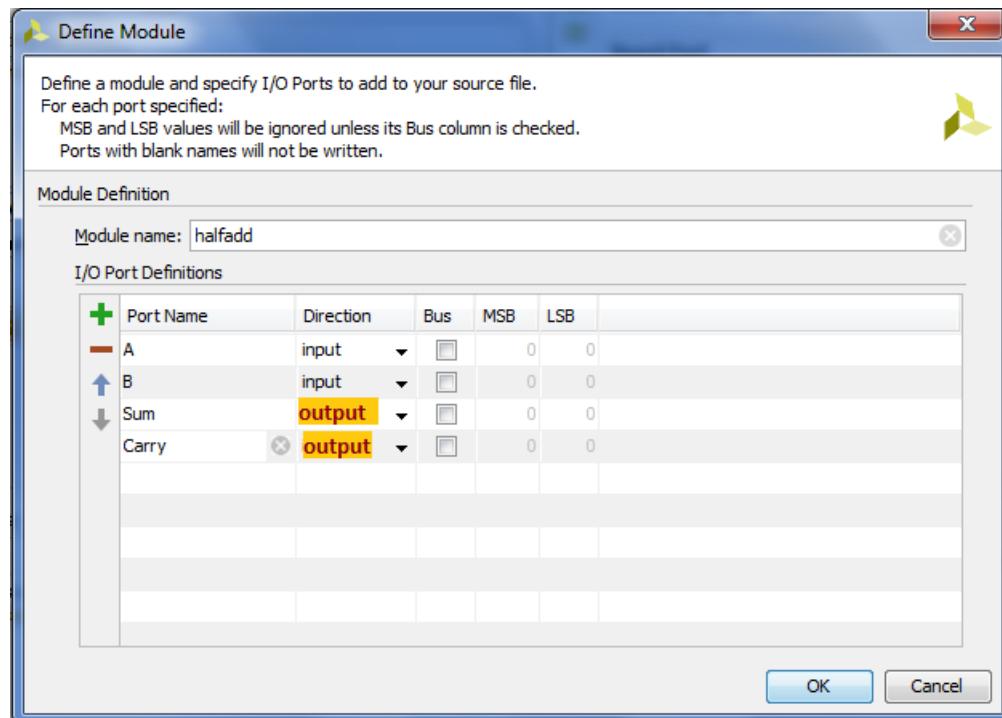


Figure 11: Mention the IOs of the verilog files created

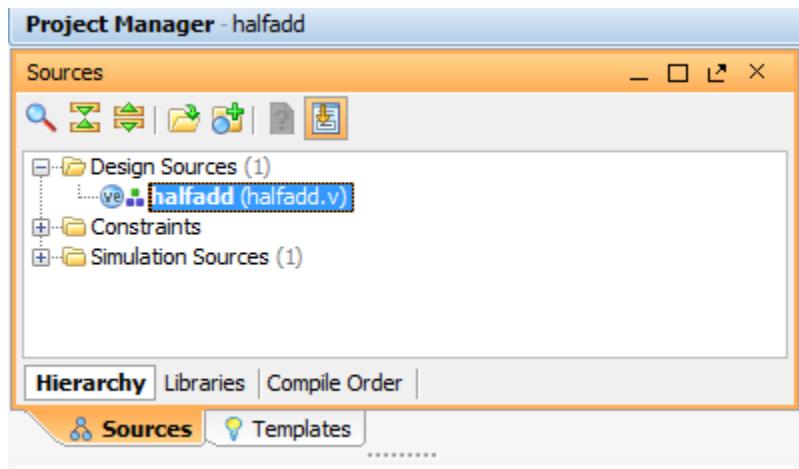


Figure 12: You can see the files added in **Sources** window. Now open the files and code the required logic.

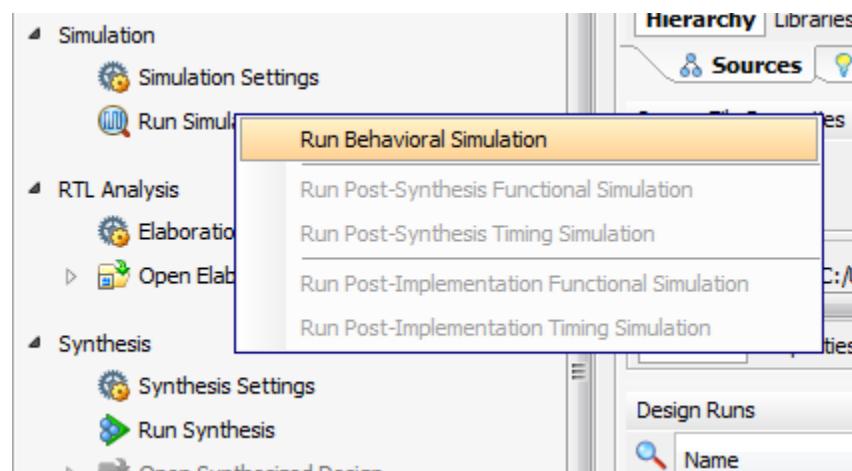


Figure 13: After writing the Verilog code, run the **Simulation**.

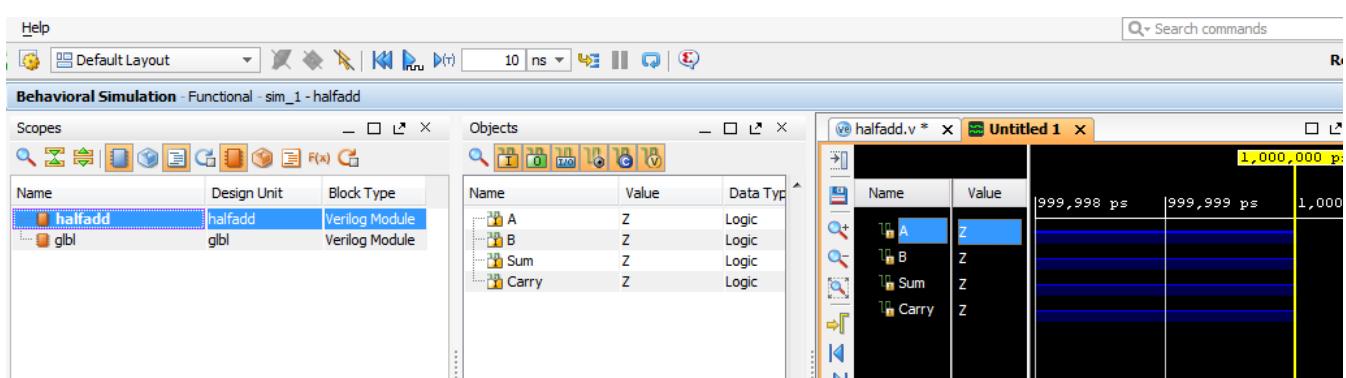


Figure 14: Behavioral Simulation will open with the following windows.

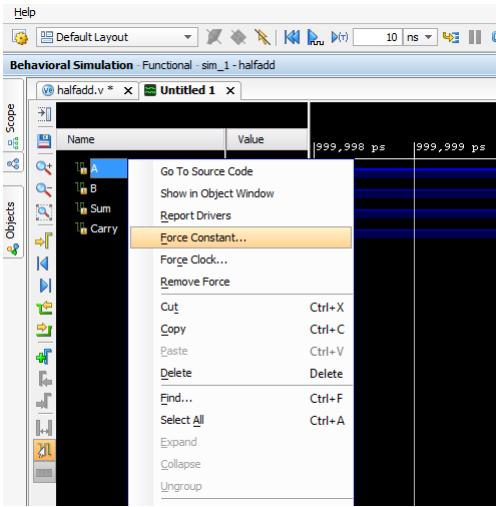


Figure 15: Force constant or clocks to the Inputs to generate various test sequences.

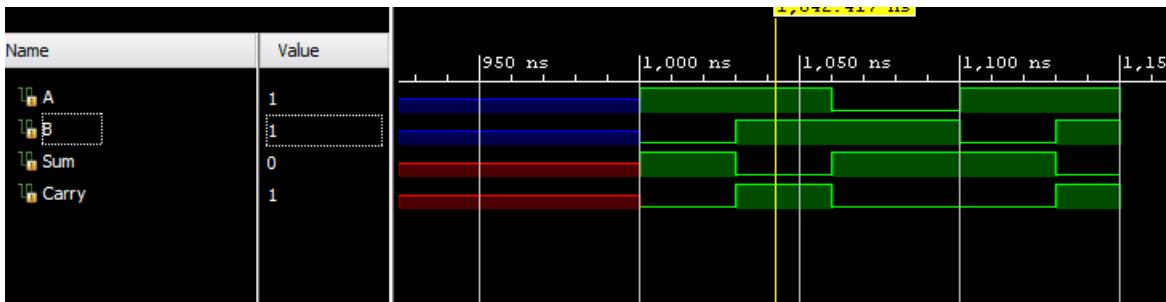


Figure 16: Verify the outputs generated for corresponding Inputs

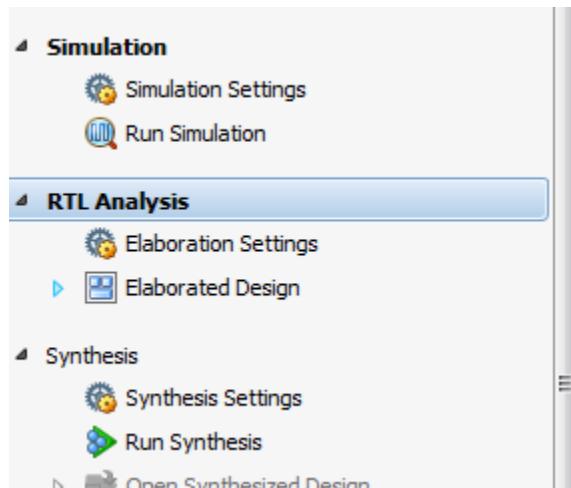


Figure 17: Now check the Elaborated design from RTL analysis

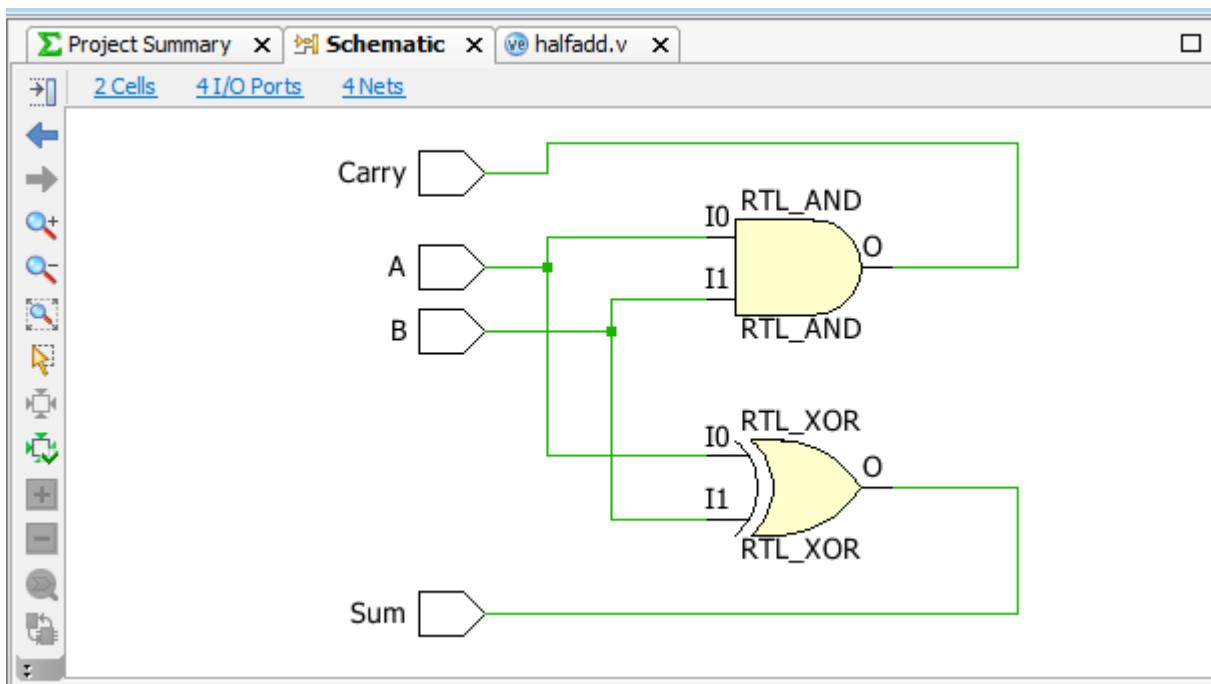


Figure 18: Schematic will be displayed.

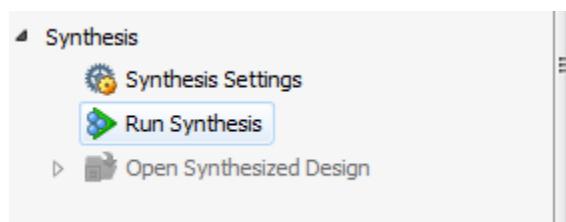


Figure 19: Now run the Synthesis



Figure 20: After successfull synthesis, run the Implementation

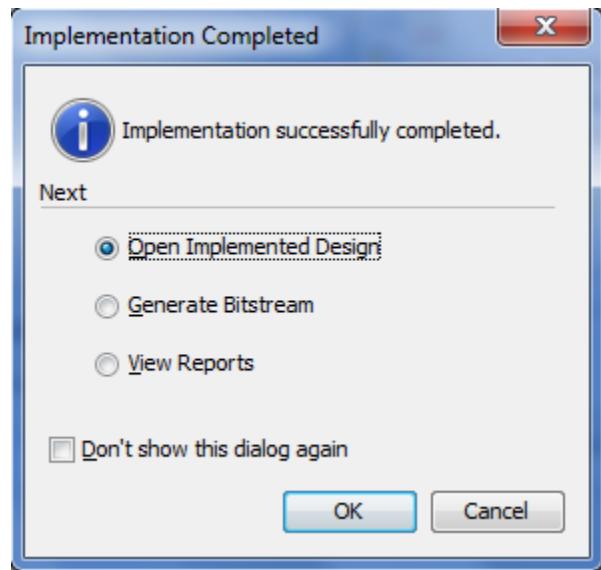


Figure 21: Open the Implemented Design after successful Implementation

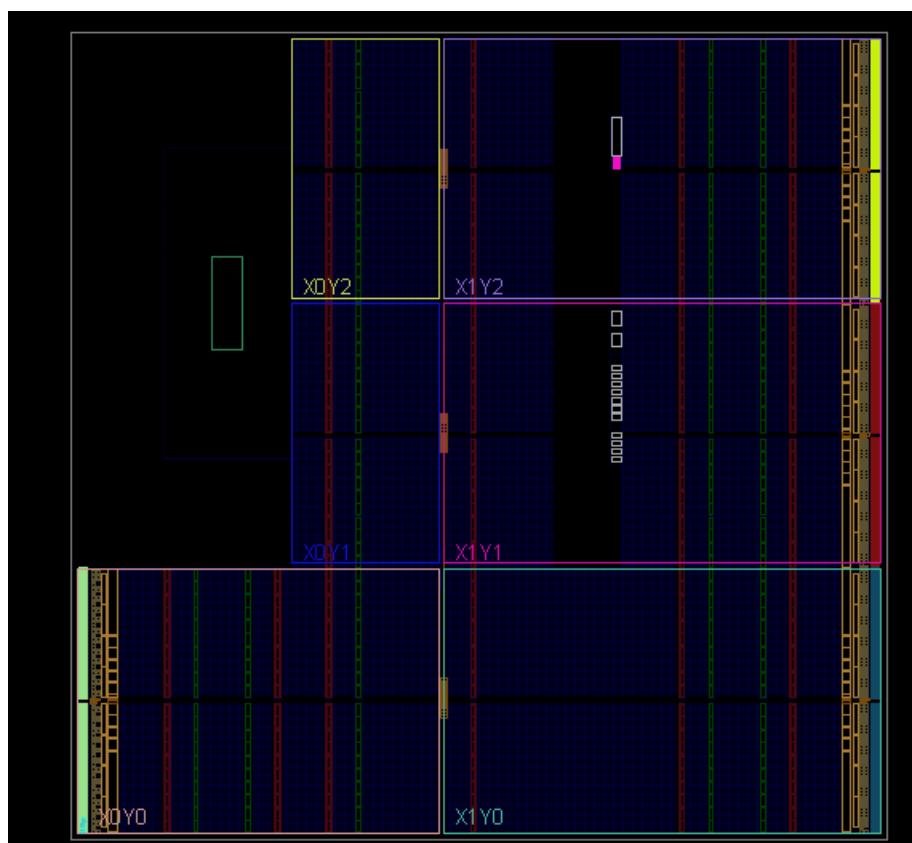


Figure 22: The Zedboard architecture

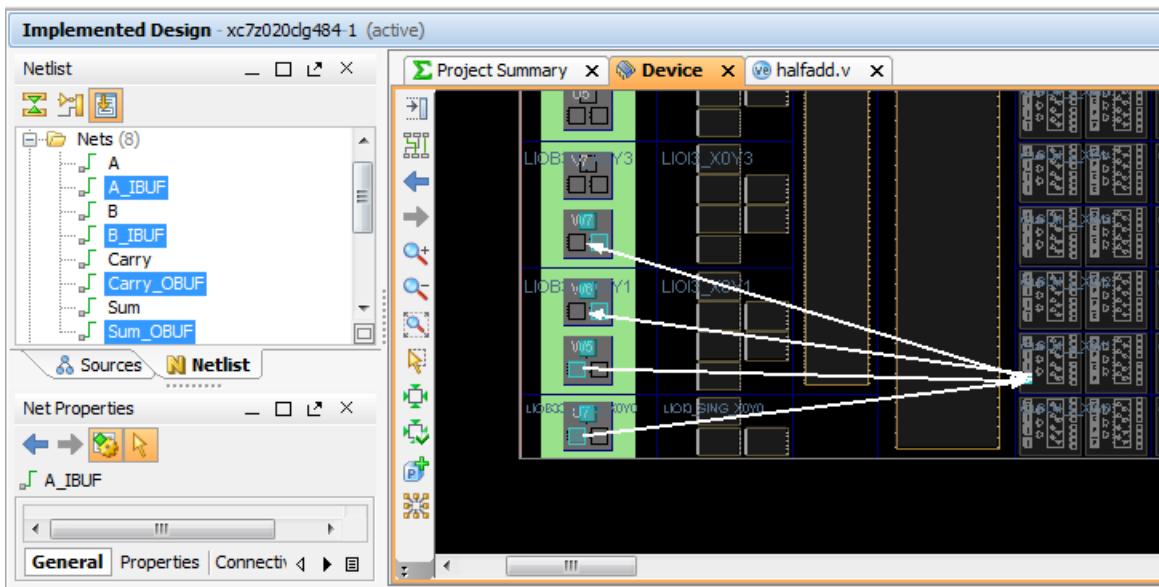


Figure 23: Zoom In to see the implemented Logic, and select different Nets to visualise the connections.

Resource	Utilization	Available	Utilization %
LUT	1	53200	0.01
IO	4	200	2.00

Figure 24: Project Summary gives details on resource utilization, timing and other details.

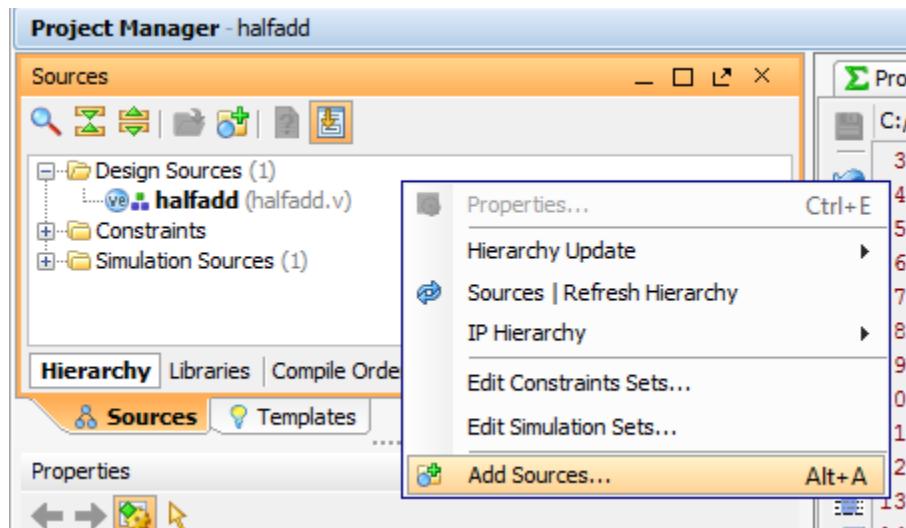


Figure 25: After successful implementation add .xdc file into the project.

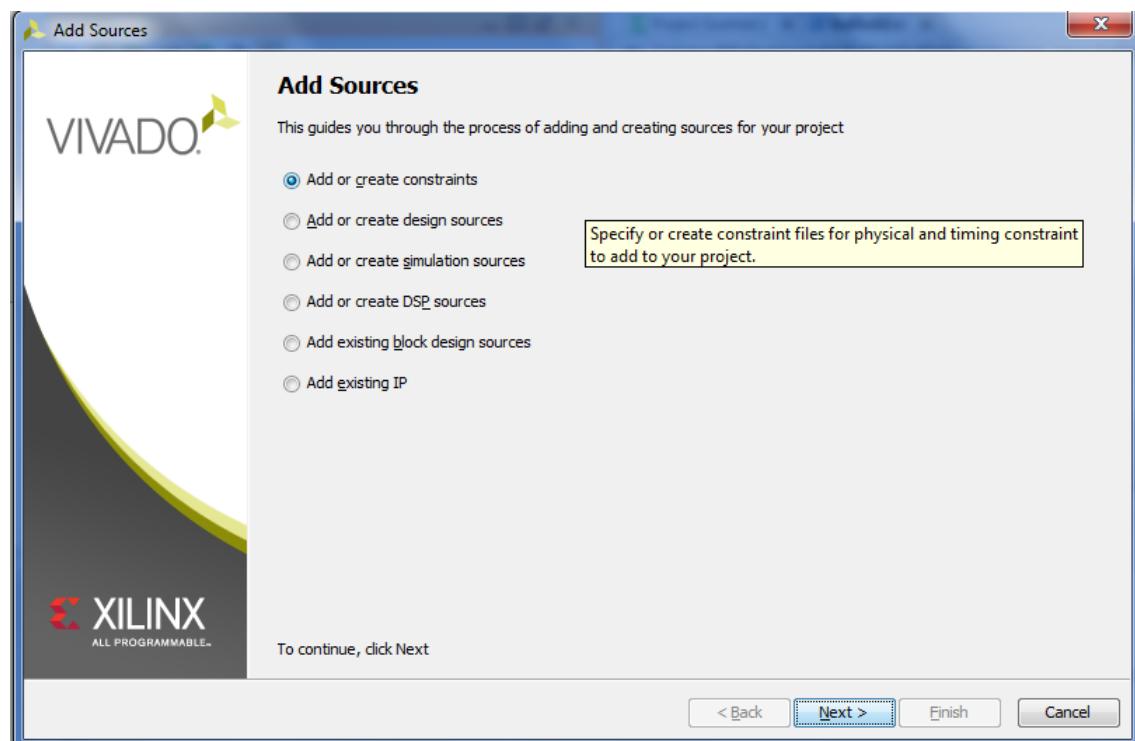


Figure 26: Click on **Add or create constraints**.

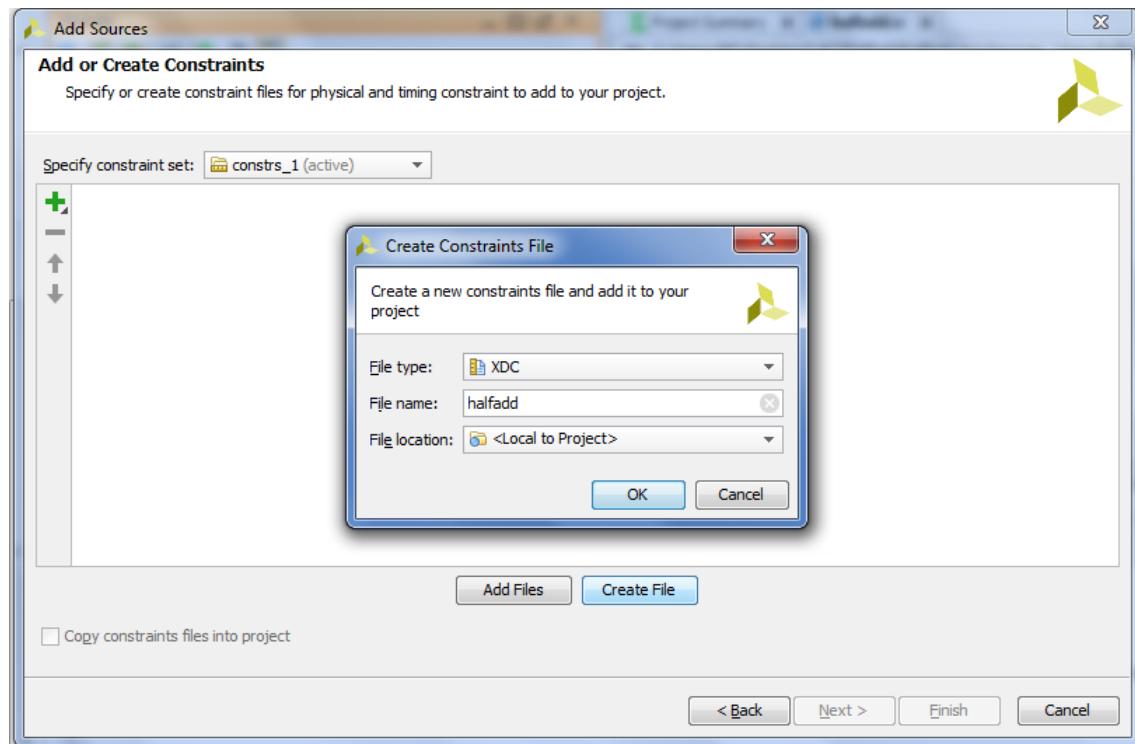


Figure 27: **Create File** and give a name to your .xdc file

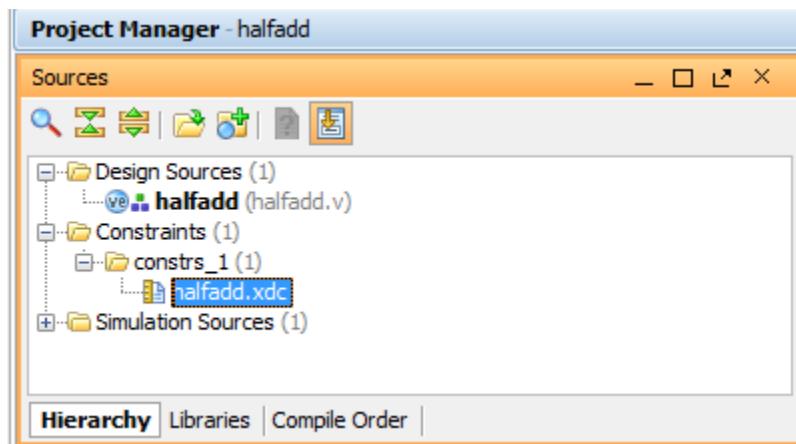


Figure 28: .xdc file shows up in the **Sources** window under **Constraints**. Now open the file and write the constraints to make connections between the IOs and the peripherals on the board.

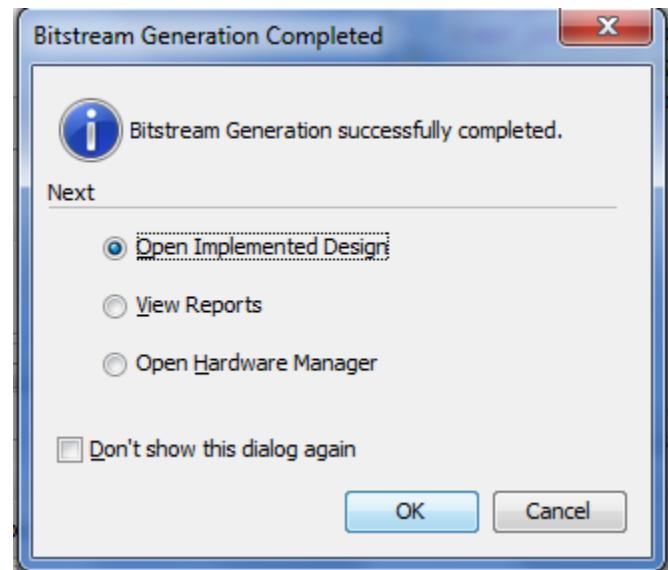


Figure 29: After completing the .xdc file, click on **Generate Bitstream**. Above window will appear on successful Bitstream Generation. Now power on the board and connect it to the PC/Laptop. Click on **Open Hardware Manager**.

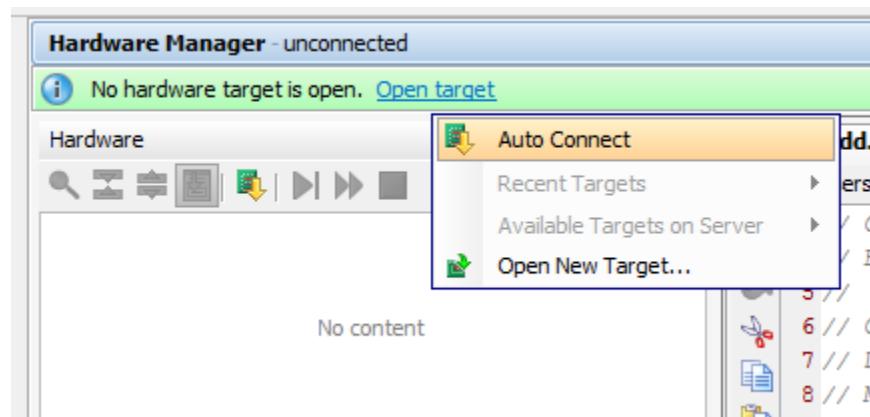


Figure 30: Click on **Open Target** and **Auto Connect**.

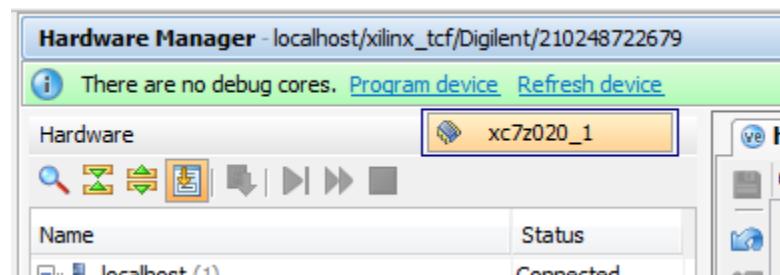


Figure 31: Click on **Program device** and select the FPGA device.

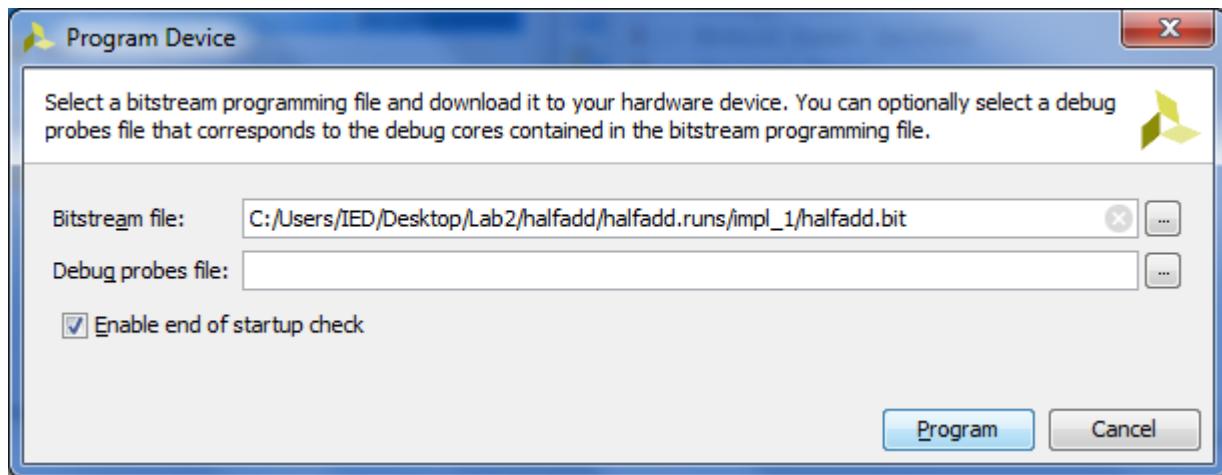


Figure 32: Select the bit file to be used to program the FPGA. Click on **Program**. The FPGA will be successfully programmed. Verify your design using on-board switches and LEDs.

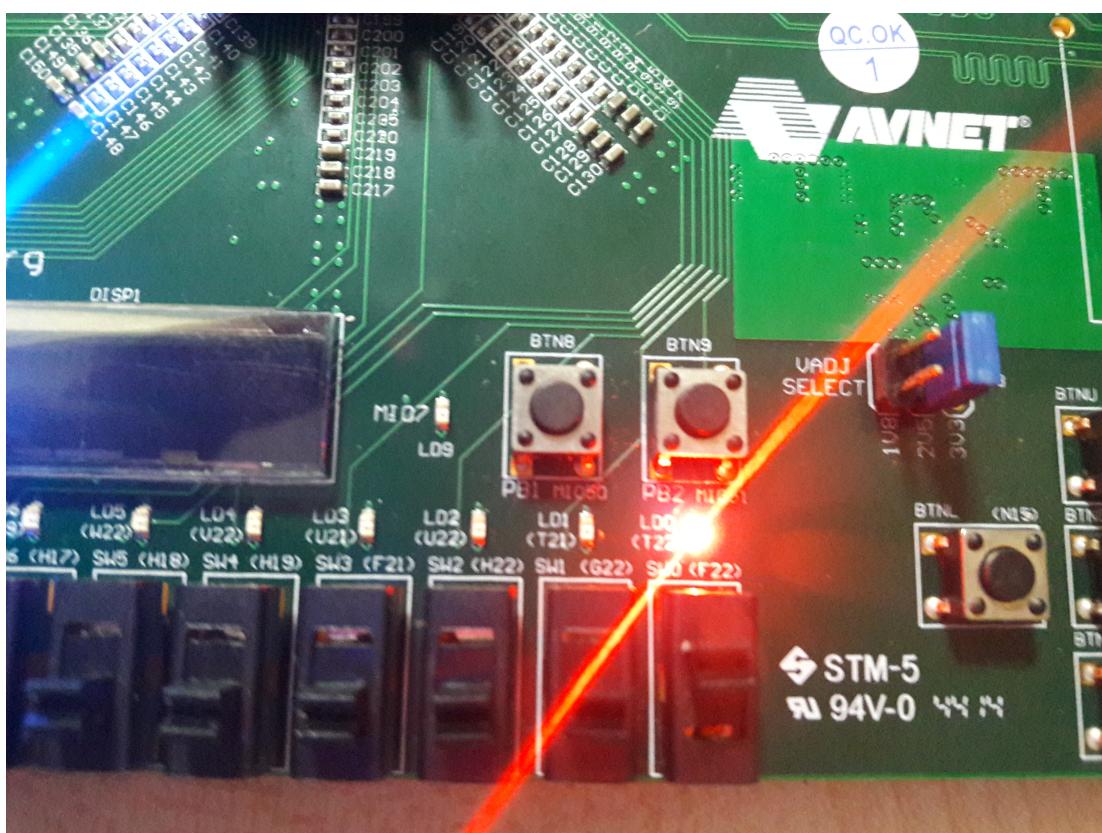


Figure 33: Result for $A(SW0)=1$, $B(SW1)=0$; Sum(LD0)=1, Carry(LD1)=0

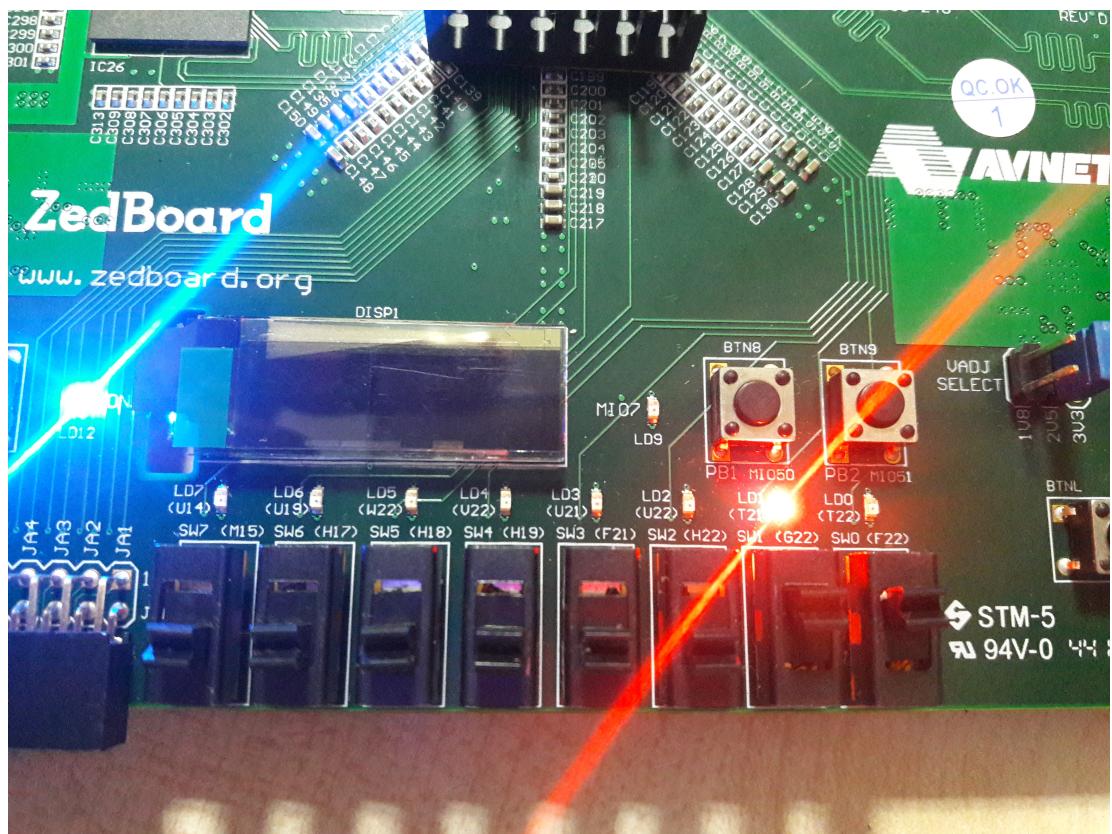


Figure 34: Result for $A(SW0)=1$, $B(SW1)=1$; $\text{Sum}(LD0)=0$, $\text{Carry}(LD1)=1$