Experiment 01

Objective: Implement a half-adder on the PYNQ Z2 FPGA board using Verilog through the Vivado Design Suite.

Apparatus: Vivado design software, PYNQ Z2 board, USB cable

Theory: PYNQ is an open-source project from Xilinx that makes it easy to design embedded systems with Zynq Systems on Chips (SoCs). Using the Python language and libraries, designers can exploit the benefits of programmable logic and microprocessors in Zynq to build more capable and exciting embedded systems. PYNQ-Z2 is a FPGA development board based on ZYNQ XC7Z020 FPGA, intensively designed to support PYNQ, a new open-sources framework that enables embedded programmers to explore the possibilities of xilinx ZYNQ SoCs without having to design programming logic circuits. Benefiting from programmable logic and advanced ARM processor in ZYNQ, designers can build up more powerful embedded systems with it. Besides, the SoCs can be programmed in Python and the code can be developed and tested directly on the PYNQ-Z2. The programmable logic circuits are imported as hardware libraries and programmed through APIs in basically the same way that the software libraries are imported and programmed. PYNQ-Z2 board integrates Ethernet, HDMI Input/Output, MIC Input, Audio Output, Arduino interface, Raspberry Pi interface, 2 Pmod, user LED, push-button and switch. It is designed to be easily extensible with Pmod, Arduino, and peripherals, as well as general purpose GPIO pins.

BOARD LAYOUT

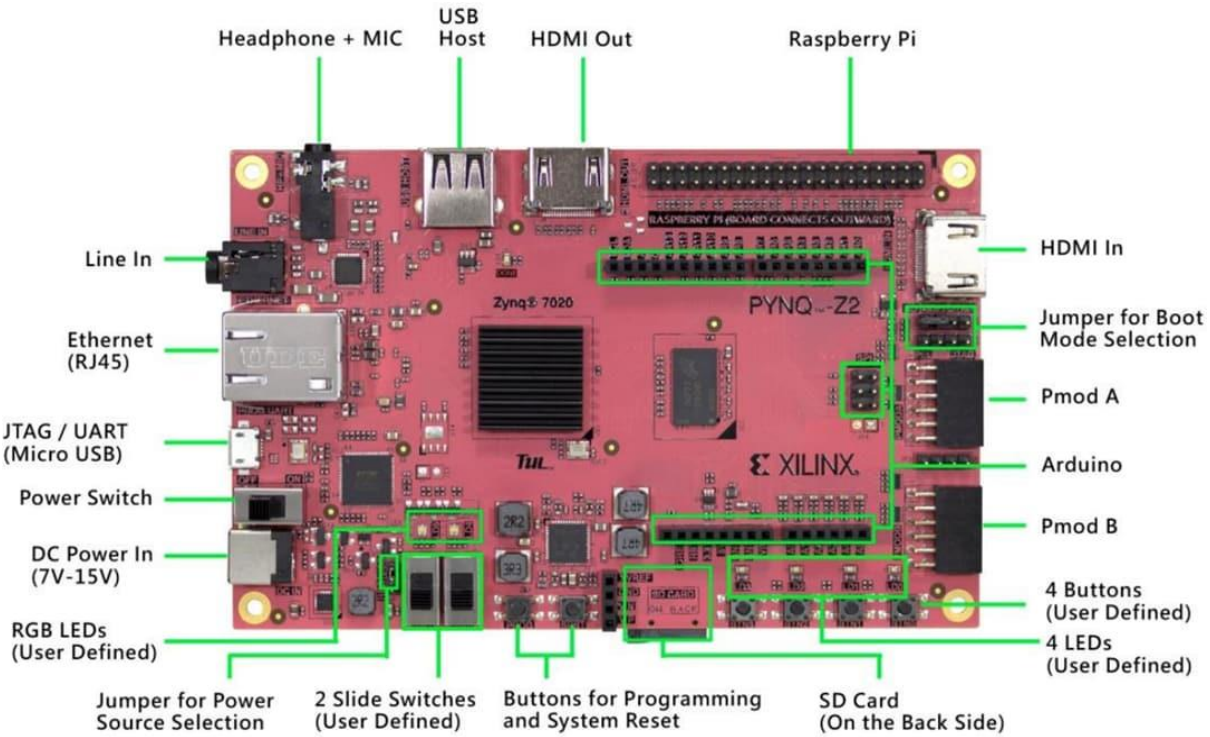

Figure 1

Vivado Design Suite: Vivado Design Suite is a software suite produced by Xilinx for synthesis and analysis of hardware description language (HDL) designs, superseding Xilinx ISE with additional features for system on a chip development and high-level synthesis. Like the later versions of ISE, Vivado includes the in-built logic simulator.

Procedure: First Download the Board file and Master XDC file using following link:

https://www.tulembedded.com/FPGA/ProductsPYNQ-Z2.html

After download paste these **board file** in the following path of your laptop:

C:\Xilinx\Vivado\2018.3\data\boards\board_files

Create a Vivado Project using IDE Launch Vivado and create an empty project targeting PYNQ-Z2 board, selecting Verilog as a target language.

1. Open Vivado by selecting **Start > Vivado 2018.3**
2. Click **Create New Project** to start the wizard. You will see *Create a New Vivado Project* dialog box. Click **Next**.
3. Click the Browse button of the *Project location* field of the **New Project** form, browse to **C:\xup\fpga_flow\2021_2_zynq_labs**, and click **Select**.
4. Enter **lab1** in the *Project name* field. Make sure that the *Create Project Subdirectory* box is checked. Click **Next**.
5. Select **RTL Project** option in the *Project Type* form and click **next**. You may check the box on 'Do not specify sources at this time'
6. In the *Default Part* form, use the **Parts** option and various drop-down fields of the **Filter** section. Select the **PYNQ Z2 from the board.**
7. Click **Finish** to create the Vivado project.

Create Design Source

In the *Sources* pane, right click and **create** a new Design source file, eg. **lab1.v**.

Double-click the **lab1.v** entry to open the file in text mode. **Write the Verilog module for half adder here.** (You may also include testbench in the same code or create a separate Verilog file under Simulation sources.)

Upload and update the Constraint File

In the *Sources* pane, expand the *Constraints* folder and double-click the **pynq_z2.xdc** entry to open the file in text mode.

Right click on the Constraints folder and add a new Constraint. Instead of creating a new file, add the path to the PYNQ XDC file.

Note that the Xilinx Design Constraints file assigns the physical I/O locations on FPGA to the switches and LEDs located on the board. This information can be obtained either through the board's schematic or the board's user guide.

Perform RTL analysis on the source file

Expand the *Open Elaborated Design* entry under the *RTL Analysis* tasks of the *Flow Navigator* pane and click on **Schematic**.

The model (design) will be elaborated, and a logic view of the design is displayed. Download/take a screenshot of this design.

Synthesize the Design

Synthesize the design with the Vivado synthesis tool and analyze the Project Summary output.

1. Click on **Run Synthesis** under the *SYNTHESIS* tasks of the *Flow Navigator* pane.

   The synthesis process will be run on the lab1.v file (and all its hierarchical files if they exist). When the process is completed, a *Synthesis Completed* dialog box with three options will be displayed.

2. Select the *Open Synthesized Design* option and click **OK** as we want to look at the synthesis output before progressing to the implementation stage.

   Click **Yes** to close the elaborated design if the dialog box is displayed.

Implement the Design

Implement the design with the Vivado Implementation Defaults settings and analyze the Project Summary output.

1. Click on **Run Implementation** under the *Implementation* tasks of the *Flow Navigator* pane. The implementation process will be run on the synthesized design. When the process is completed, an *Implementation Completed* dialog box with three options will be displayed.

2. *Select Open implemented design and click OK as we want to look at the implemented design in a Device view tab.*
3. Click **yes,** if prompted, to close the synthesized design. The implemented design will be opened.

Generate the Bitstream and Verify Functionality

Connect the board and power it ON. Generate the bitstream, open a hardware session, and program the FPGA.

1. Make sure that the Micro-USB cable is connected to the JTAG PROG connector.

2. PYNQ-Z2 can be powered through USB power via the JTAG PROG. Make sure that the board is set to use USB power.

3. Power **ON** the board.

4. Click on the **Generate Bitstream** entry under the *PROGRAM AND DEBUG* tasks of the *Flow Navigator* pane.

   The bitstream generation process will be run on the implemented design. When the process is completed a *Bitstream Generation Completed* dialog box with three options will be displayed.

   This process will have generated a **lab1.bit** file under the **impl_1** directory in the **lab1.runs** directory.

5. Select the *Open Hardware Manager* option and click **OK**.

   The Hardware Manager window will open indicating "unconnected" status.

6. Click on the **Open target** link. From the dropdown menu, click **Auto Connect.**

   The Hardware Session status changes from Unconnected to the server name and the device is highlighted. Also notice that the Status indicates that it is not programmed.

   Select the device and verify that the lab1.bit is selected as the programming file in the General tab.

7. Click on the *Program device* link in the green information bar to program the target FPGA device. Another way is to right click on the device and select *Program Device*.

9. Click **Program** to program the FPGA.

   The DONE LED will light when the device is programmed. You may see some other LEDs lit depending on switch positions.

10. Verify the functionality by flipping the switches and observing the output on the LEDs (Refer to the earlier logic diagram).

11. When satisfied, power **OFF** the board.

12. Close the hardware session by selecting **File > Close Hardware Manager.**

13. Click **OK** to close the session.

14. Close the **Vivado** program by selecting **File > Exit** and click **OK.**

Download the Lab Report Template and create a submission containing the report and a video of the working.

Online Reference

R. Chouhan, *Introduction to Vivado Design Suite* (**https://youtu.be/A54x7T8pnHI**)