Laboratory Session 6

Basic Interfaces: LEDs, Push Buttons, and DIP Switches

Objective

- Implement Hardware Descriptions using VHDL
- Analyze Hardware Implementations using Quartus and ModelSim
- Interface FPGA Boards

Software Packages

- Altera Quartus II
- ModelSim Software

Hardware Devices

- DE2i-150 FPGA Board

1. Introduction

The Altera DE2-70 development board is equipped with Altera's Cyclone® II 2C70 Field Programmable Gate Array (FPGA). The Altera DE2 board contains multimedia, storage, networking interfaces, and large memory components (See Figure 1). The board offers a rich set of features that make it suitable to be used in a laboratory environment for a variety of design projects, as well as for the development of sophisticated digital systems.

Figure 1: DE2i-150 Application Board

2. Samples

2.1.Full Adder Using Pushbuttons

Configure the FPGA to work as a 1-bit Full-Adder based on the implementation from Laboratory Session 1. Apply the following:

- Create a new folder (eg. Full_Adder).
- Create a project (eg. Full_Adder) by selecting the folder created and providing a name for the project and top-level entity, then Next.
- Add the behavioral Full_Adder VHDL file or copy-paste the file to the same project folder and add it to the project.
- In “Family & Device Settings Window”, select for family: Cyclone IV and from Available devices: EP4CGX150DF31C7.

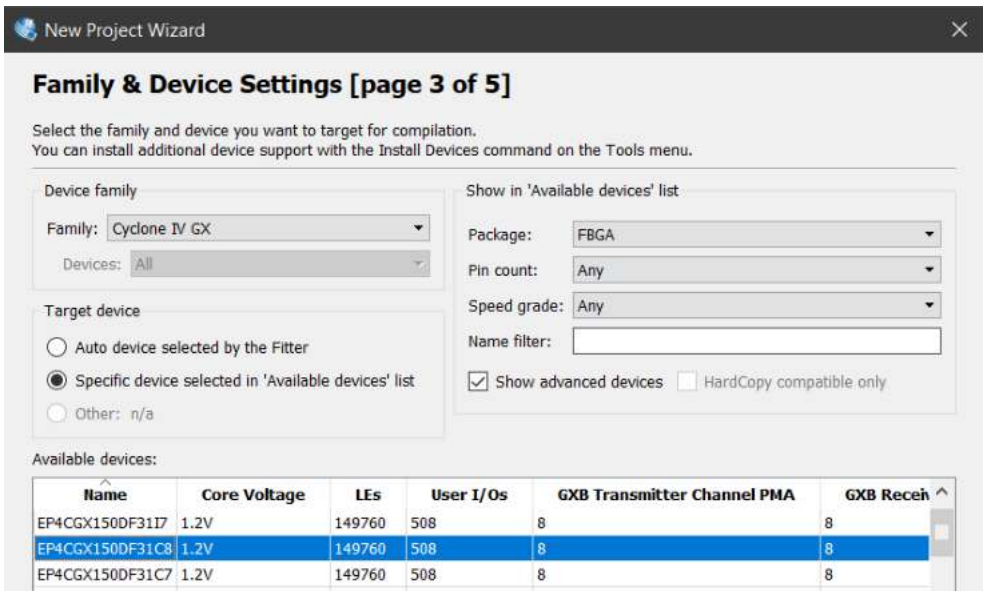

Figure 2: Family & Device Setting Window

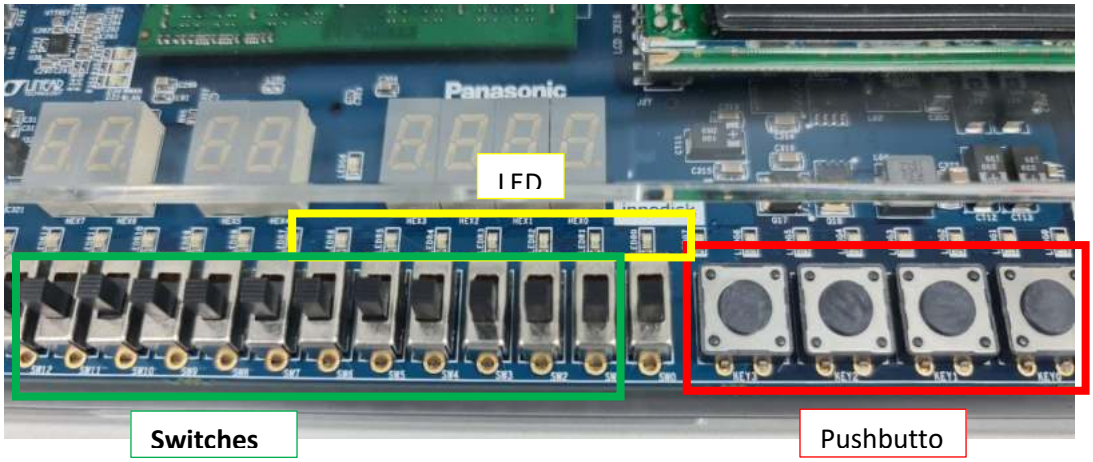

Figure 3: FPGA switches, pushbuttons, and LEDs

- Compile the code to make sure it is working correctly and to generate all the necessary files.
- To use the Full-Adder on the FPGA board, use 3 push buttons for the inputs a, b and cin, and 2 LEDs for the outputs s and cout.
- In the toolbar, open the Assignments menu and select Pin Planner.
- The Pin Planner window will appear, at the bottom pane (All Pins) all the input and output pins of the top-level entity will be listed.
- For Node Name "a", type PIN_AA26 under location and press enter.
- Repeat the same for the rest of the pins
  - b: PIN_AE25
  - cin: PIN_AF30
  - cout: PIN_T23
  - s: PIN_T24

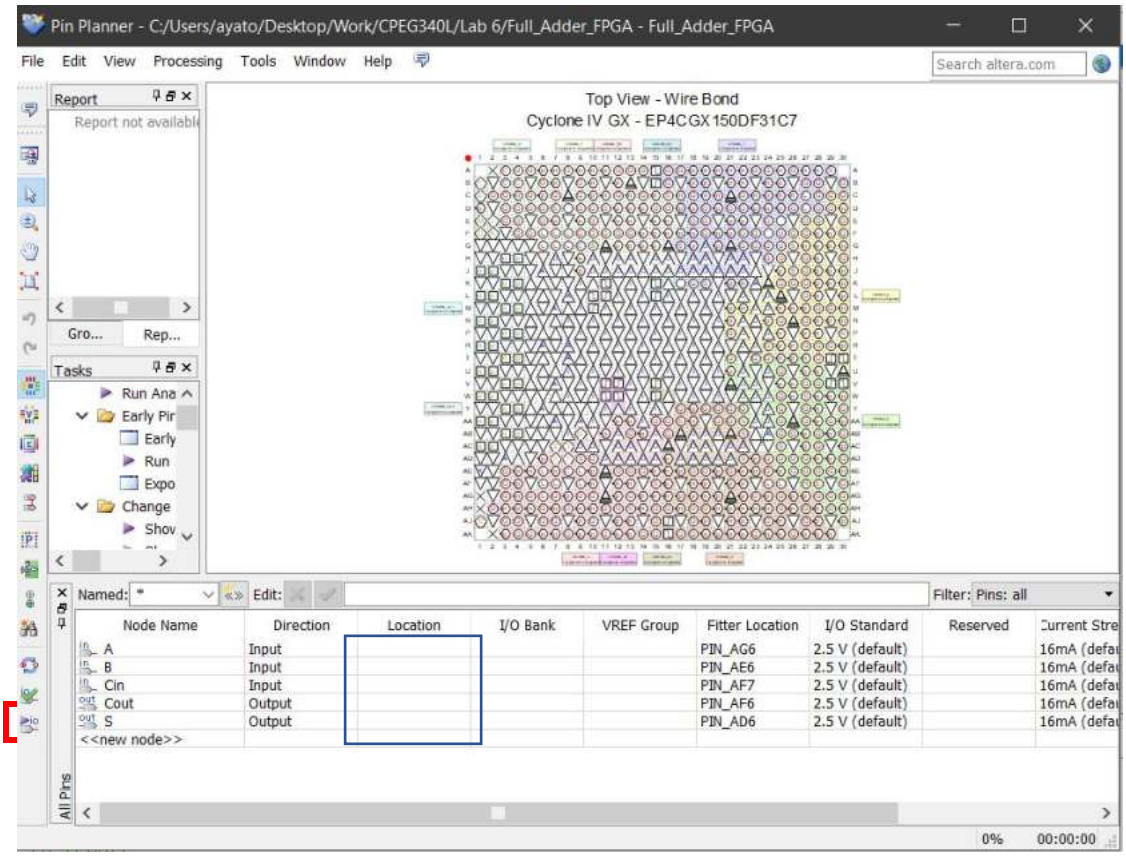

Figure 2: Pin Planner Window

- This assignment allows you to use the pushbuttons labeled **KEY0**, **KEY1** and **KEY3** for "a", "b" and "cin" respectively and the red LEDs **LEDR0** and **LEDR1** for "s" and "cout" respectively. To understand where these values are available, please check pages 32 and33 of the System Manual available on Moodle.
- Click on Start I/O Assignment Analysis on the left toolbar and wait until the analysis is done. Then click OK.

- Go to File on the Pin Planner window and select Export. The Export Dialog will appear select a location to save the file preferably inside the project folder.
- Close the Pin Planner window.
- Go to Assignment > Import Assignments…, and select the exported file with .csv extension
- Compile the project again to apply the pin assignments

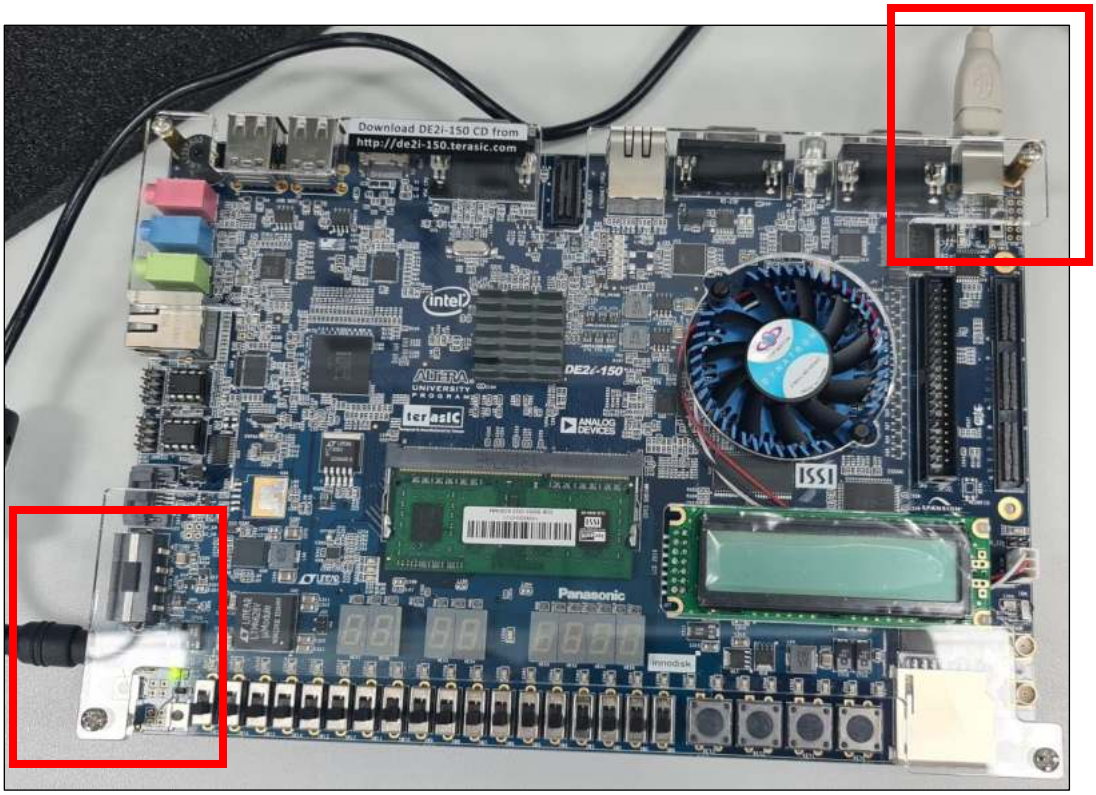

Figure 3: Power and USB Connections on the board

- Connected the power cable to the FPGA and the USB cable to the USB Blaster port on the FPGA and to any USB port on the PC
- Power ON the FPGA
- Click on Tools menu and select Programmer
- The Programmer menu will appear, click on Hardware Setup.
- The Hardware Setup window will appear. In the drop list, next to Currently selected hardware, select USB Blaster and close the window
- Click on Start and wait until the progress bar at the top right becomes 100%
- Test the Full Adder on the FPGA.

Note: The pushbutton provides logic 1 when not pressed and logic 0 when pressed.

Note: When the FPGA is turned OFF, the configuration is cleared.

2.2. Parity Generator Using Switches

Design, implement and verify an odd parity generator. The generator takes a 4-bit binary value entered using switches. The parity generator outputs the odd parity bit for the 4-bit input using an LED. After Writing the code of the Parity Generator, make sure to follow the below details to make sure to assign switches to your input and

Open the Pin Planner window.

1. On Moodle, allocate the DE2i-150 manual, if it is not available you can find it online.
2. Open the manual and go to section 3.4 “Using LEDs”.
3. Scroll down until you find a table listing the signal name and the FPGA pin no.
4. Use the following switches SW0, SW1, SW2, and SW3 for a, b, c, and d respectively.
5. Repeat the steps from the previous exercise. Note that you can export the assignments with a different name to preserve the previous assignment.

Note: The switch provides logic 1 when it is position towards the LEDs and 0 when positioned towards the edge of the board.

3. Practice Exercises

3.1. Full Adder using Switches

Use the same project as in the previous Sample Exercise but change the inputs to be switches as follows:

1. On Moodle, allocate the DE2i-150 manual, if it is not available you can find it online.
2. Open the manual and go to section 3.4 “Using LEDs”.
3. Scroll down until you find a table listing the signal name and the FPGA pin no.
4. Use the following switches SW0, SW1, and SW2 for a, b, and cin respectively.
5. Repeat the steps from the previous exercise. Note that you can export the assignments with a different name to preserve the previous assignment.

3.2. BCD Counter

Design, implement and verify a BCD Counter. The Counter should count from 0 to 9 and repeats. Use a pushbutton to increment the counter by 1, a switch to reset the value to 0 and 4 LEDS to display the current count.

4. Deliverables

Practical demonstration of the sample exercises will be requested during the lab. Keep in mind that a combined report that presents the analysis and testing of the sample exercises as well as the practice exercises of lab 1 and 2 will be requested, so maintain your project file for the snapshots of the work done.