ELE307 Spring 2009 LAB1

What's in this lab:

VHDL	IEEE library and packages, entity, architecture, concurrent assignments
EDA	New project, VHDL editor, pin assignments, compilation, vector waveform editor, simulation, programmer, archive project.
FPGA	LEDs, pushbuttons and switches
RTL	bit-wise logic operators, wire grouping

Exercise (Creating a new project)

Students are to take turn taking the lead. Labs 1 to 8 are individually report but the entire lab should be participated by both students in the team. When you are in the lead, use your own ELE account. You will use your ELE account's email address for all correspondences.

Step 1: Create and Configure the Project

- The main program used for the labs is electronic design automation (EDA) tool by *Altera* called *Quartus II*. The current version is 8.0. To start the software, click the "Start (K Menu icon) → Run Command…". Type "**quartus**" in the window and hit return. Be sure to use all lowercase letters since Linux is case sensitive. The program may take a moment to start. Alternatively, you may type "quartus" in the command line inside a Linux terminal.
- To start a new project, click "*File → New Project Wizard"*. Once the wizard window appears, click the **Next** button past the introduction. The three fields presented on the next page specify what directory, project name, and top-level entity are used. For simplicity, all three fields should specify "lab1". In the first field (topmost), add "*lab1*" so it shows /u/ugrads/yourname/lab1. Type in "*lab1*" to the second field and the third field will be filled automatically.
- **Remember to click on "Next"** not Finish yet!!! On next page, click next since there is no file from other projects that we want to import.
- On next page, select the followings:
  - Device family: "Cyclone II"
  - Target device: Specific device selected in 'Available devices' list
  - Available devices: select "EP2C35F672C6"
  - Filters: choose 'any' for all three fields.
- Skip page 4 as we do not use additional EDA tools.
- Page 5 is a summary page of your project settings. Click Finish to complete the process of creating the project.

Step 2: VHDL editor

- "File → New" and select "VHDL File" under the "Design Files" group.
- Type the following VHDL codes: (Case insensitive)

```
Library IEEE;
Use IEEE.std_logic_1164.all;
ENTITY lab1 is
        Port(
                SW: in std_logic_vector(17 downto 0);
```

```
                LEDR: out std_logic_vector(17 downto 0)
        );
END lab1;

ARCHITECTURE exercise of lab1 is
BEGIN
        LEDR(3 downto 0) <=
                SW(7 downto 4) and SW(3 downto 0) when SW(17 downto 16)="00" else
                SW(7 downto 4) or  SW(3 downto 0) when SW(17 downto 16)="01" else
                not SW(3 downto 0) when SW(17 downto 16)="10" else
                SW(7 downto 4) nand (3 downto 0 => SW(2));
END ARCHITECTURE exercise;
```

- Use "File → Save As" to save this VHDL as "lab1.vhd". Conventionally, the top-level VHDL file name should be the same as the project name.

Step 3: Functional Simulations

- "Processing → Start Compilation" will process the above VHDL code and generate a logic circuit fitted to the FPGA. Any syntax error may be identified here. The report will be displayed at the bottom window pane. The errors are shown in red. Double click on the error will lead you back to the line in the VHDL editor. When the compilation is successful, a compilation report window will show up giving you the "Flow Summary."
- "Assignments → Settings" will bring out a window of settings. On the left, select "Simulator settings". Then, on the right, make sure it is "Timing" for simulation mode. Click "OK" and leave everything else blank.
- "File → New" and choose to create a "Vector Waveform File". This will bring out the waveform file editor. You will use this editor to set up the input to the circuit.
- Double click on the left narrower field (labeled name, etc.) will bring out a small window of "Insert Node or Bus". Click on "Node Finder…" to bring out the node finder window. Make sure "*" in the field called "Named" and "Pins:all" in "Filter" and then click on "List". All the inputs and outputs declared in the VHDL entity will show up in the left window pane labeled "Nodes Found:". In this case, you will see LEDR is listed as one entry and LEDR[0] to LEDR[17] are listed separatedly. Choose LEDR and SW as they represented the entire set of LED[x] and SW[x]. Double click the entry or use the ">" key in the middle to move the selected inputs and outputs to the right window pane. Then, click "OK". Click "OK" in the "Insert Node or Bus" window.
- Remember in Vector Waveform File we have to select both inputs and outputs but we will only set values to the inputs. The outputs are selected to force the simulators to display results on the selected outputs.
- Click on the "+" radio button next to SW to expand the view. You will now see SW[0] to SW[17] inputs. In this exercise, we use only SW[17], SW[16] and SW[7] to SW[0]. Single click SW[0] to highlight it and then right click "Value → Forcing High (1)" to set the value of SW[0] to logic 1. Alternatively, you may use the button on the left menu to set the value to 1. Similarly, set the following values SW[1]=0, SW[2]=1, SW[3]=0, SW[4]=0, SW[5]=1, SW[6]=0 and SW[7]=1.
- SW[17] and SW[16] are used to select the different logic functions, so we need to provide all four possible combinations. Right click SW[16], "Value → clock…" (or the menu button on the right with a stopwatch symbol). Set "Period" to 100ns. Similarly set

SW[17]'s period to 200ns. The waveform on the right should reflect the changes immediately. Use "zoom" to change the view to see the waveform.

- Save this vector waveform file as "lab1.vwf"
- "Processing → Start Simulation" will produce a simulation report. The waveform part looks and feels exactly the same as in the waveform editor. However, remember that this is the report and if you need to modify something, you go to the waveform editor.
- Verify the results against the VHDL code description. Since this is the timing simulation, the accurate circuit delay time is accounted for, so don't expect the outputs to change instantaneously as the input changes. You will have to figure out the Boolean functions for your report. Try to do this before next step as it may confuse you.

Step 4: RTL Viewer

- "Tools → Netlist Viewers → RTL Viewer" will generate a RTL-level logic diagram.
- "Tools → Chip Planner (Floorplan and Chip Editor)" will generate a view of the inside of the FPGA. You will need to zoom in the see the seven logic elements used in this design.

Step 5: Pin Assignments

The FPGA's pins on the DE2 board have been pre-defined. During the above process, the compiler has identified the necessary input and output pins for the logic circuit. However, these pins' numbers are yet to be specified. (Note that the compilation target is the FPGA not the FPGA board)

- "Assignments → Pins" or "Assignment → Pin Planner" will bring out the "Pin Planner" window. The lower part is where the pins are assigned.
- Use the following pin numbers (locations)

Node Name	Location	Node Name	Location
LEDR[17]	PIN_AD12	SW[17]	PIN_V2
LEDR[16]	PIN_AE12	SW[16]	PIN_V1
LEDR[15]	PIN_AE13	SW[15]	PIN_U4
LEDR[14]	PIN_AF13	SW[14]	PIN_U3
LEDR[13]	PIN_AE15	SW[13]	PIN_T7
LEDR[12]	PIN_AD15	SW[12]	PIN_P2
LEDR[11]	PIN_AC14	SW[11]	PIN_P1
LEDR[10]	PIN_AA13	SW[10]	PIN_N1
LEDR[9]	PIN_Y13	SW[9]	PIN_A13
LEDR[8]	PIN_AA14	SW[8]	PIN_B13
LEDR[7]	PIN_AC21	SW[7]	PIN_C13
LEDR[6]	PIN_AD21	SW[6]	PIN_AC13
LEDR[5]	PIN_AD23	SW[5]	PIN_AD13
LEDR[4]	PIN_AD22	SW[4]	PIN_AF14
LEDR[3]	PIN_AC22	SW[3]	PIN_AE14
LEDR[2]	PIN_AB21	SW[2]	PIN_P25
LEDR[1]	PIN_AF23	SW[1]	PIN_N26
LEDR[0]	PIN_AE23	SW[0]	PIN_N25

- **Important:** Compile the circuit again so the pin assignments can take place.

Step 6: Programming the DE2 board

- "Tools → Programmer" will bring out the programmer window.
- Make sure it shows "USB" , Mode is "JTAG".
- "lab1.sof" and check "Program/Config".
- Make sure DE2's power is on. Click "Start".
- Use the switches on the DE2 board to verify that the logic circuit has been "programmed" to the FPGA.

Step 7: Archive Project

The Quartus II project can be archived and later restore. This is a very useful function for transporting project between user accounts or between computers. For example, you can archive the project in the laboratory and bring it home and restore it on your home PC with Quartus II Web edition.

- "Project → Archive Project…" will bring out the archive project window.
- Make sure the archive name is "lab1" and then click "OK".
- In the project directory, you will find a file name "lab1.qar". This is the archived project. DO NOT use other program to open the file. Always use Quartus II to do the archive and restore the project.

Step 8: TA verification

Demonstrate to TA the programmed DE2 board. Be sure the keep notes and the project file for writing the report.

Assignment:

Each lab will contain an exercise part and an assignment part. Create a new project calls "lab1a" for this assignment. It is always a good idea to create a different project in a different directory (folder) for different sections of a lab. You may open the "lab1.vhd" file in the lab1 directory and use "save as" to save it as "lab1a.vhd" in the new "lab1a" directory. Rename the entity name to "lab1a".

Design a logic circuit that receives the four pushbuttons, key(3 downto 0), as inputs, as well as the 18 switches as in the above exercise. The outputs LEDR(7 downto 0) will display SW(15 downto 8) *and* SW(7 downto 0) when key(0) is pushed, SW(15 downto 8) *or* SW(7 downto 0) when key(1), SW(15 downto 8) *nand* SW(7 downto 0) when key(2), and SW(15 downto 8) *nor* SW(7 downto 0) when key(3). The new pin locations are:

KEY[0]=PIN_G26, KEY[1]=PIN_N23, KEY[2]=PIN_P23, and KEY[3]=PIN_W26.

Go through the same process as in the exercise. Remember that you need to compile once first to identify the pins, and then assign pin locations, and finally compile again. Show to TA after you've programmed the DE2 board and first verify the correctness yourself.

Lab 1 Report:

Title page: Lab title, section number, your name, student ID and email address. This is an individual report submit by the leader.

The body of your report should include a one paragraph introduction. Descriptions of your experiences from the exercise parts which associated with the items required below. The following required items should be embedded with the text explanation not aggregated at the end of your report. Approaches and thinking behind your solution to the assignment. Lesson learned from this lab.

From the Exercise Part:

1. Derived the Boolean functions described by the VHDL code in the exercise part.
2. Draw a logic diagram using logic gates and Multiplexer (as a logic symbol).
3. Print out of the Simulation report Waveform part.
4. Print out of the RTL.

From the Assignment part:

5. VHDL listing.
6. Print out of the Simulation report Waveform part.
7. Print out of the RTL.

Also, send the followings to TA:

1. Lab1.qar
2. Lab1a.qar

The Laboratory report is always due at the beginning of the next lab session.