Laboratory Exercise 2
ECE 441/541

Due February 1, 2016.

Part I

The purpose of Part I is, using Aldec as a front-end to Quartus, to learn how to connect simple input and output devices to an FPGA chip and implement a circuit that uses these devices. We will use the switches $SW_{0..17}$ on the DE2-series board as inputs to the circuit. We will use light emitting diodes (LEDs) and 7-segment displays as output devices.

The DE2-series board provides 18 toggle switches, called $SW_{0..17}$, that can be used as inputs to a circuit, and 18 red lights, called $LEDR_{17..0}$, that can be used to display output values. Figure 1 shows a simple VHDL entity that uses these switches and shows their states on the LEDs. Since there are 18 switches and lights it is convenient to represent them as arrays in the VHDL code, as shown. We have used a single assignment statement for all 18 LEDR outputs, which is equivalent to the individual assignments

```
LEDR(17) <= SW(17);
LEDR(16) <= SW(16);
: : :
LEDR(0) <= SW(0);
```

The DE2-series board has hardwired connections between its FPGA chip and the switches and lights. To use $SW_{0..17}$ and $LEDR_{0..17}$ it is necessary to include the correct pin assignments, which are given in the DE2-series User Manual. Either on the CD that accompanies each board or from the Altera University program web site, you can include a file with the "qsf" extension that makes the correct assignments. Using the qsf file is preferable because you do not have to recreate pin assignments every time you create a new project. More specifically the file is

- DE2_115.qsf for the DE2-115 board

The procedure for making pin assignments is described in the tutorial Aldec-Quartus tutorial.

It is important to realize that the pin assignments in the .qsf file are useful only if the pin names given in the file are exactly the same as the port names used in your VHDL entity. The file uses the names SW[0] ... SW[17] and LEDR[0] ... LEDR[17] for the switches and lights, which is the reason we used these names in Figure 3 (note that the Quartus II software uses [ ] square brackets for array elements, while the VHDL syntax uses ( ) parantheses).

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
-- Simple module that connects the SW switches to the LEDR lights
ENTITY part1 IS
  PORT ( SW : IN STD_LOGIC_VECTOR(0 TO 17); -- switches
         KEY : IN STD_LOGIC_VECTOR(0 TO 3); -- push buttons
         LEDR : OUT STD_LOGIC_VECTOR(0 TO 17); -- red LEDs
         LEDG : OUT STD_LOGIC_VECTOR(0 TO 3)); -- green LEDs
);
END part1;
ARCHITECTURE Behavior OF part1 IS
```

```
BEGIN
  LEDR <= SW;
  LEDG <= KEY;
END Behavior;
```

Figure 3. VHDL code that uses the DE2-series board switches and lights.

Perform the following steps to implement a circuit corresponding to the code in Figure 3 on the DE2-series board.

1. Create a new Aldec project for your circuit and set up the design flow to use Quartus. As a reminder, you need to select the appropriate FPGA for the DE2 board that you have:
   - Cyclone IV EP4CE115F29C7 for the DE2-115
2. Name your design part1 and your VHDL file part1.vhd
3. Copy the VHDL code from Figure 3 above into the new file and compile.
4. Select the "design flow" tab and do the following:
   a. Select/make part1 as your top level entity
   b. Select Quartus CycloneIVE as the family and EP4CE115F29C as the device. You should also set the speed grade to 7
   c. Import the pin assignment file DE2_115.qsf which you should find in the Dropbox labs folder
   d. Select the Pins tab. Set all "Dual-Purpose Pins" to "Use as regular IO"
   e. Close the options dialog and click on the "synthesis & implementation" button. This calls up Quartus to compile your model
5. Download the part1.sof into the FPGA chip. You can find this in the synthesis folder in your design folder.
6. Test the functionality of the circuit by toggling the switches and observing the LEDs.

Include in your lab report the utilization of the various FPGA resources.

Part II

In Part II, you are going to create a circuit that adds the quantities provided on the switches and displays them on the hex display. Operand 1 will be input on SW(15 downto 8), and Operand 2 will be input on SW(7 downto 0). Your circuit should display each of the operands on the paired hex displays and the sum on the group of four hex displays. Find the details on the hex displays in the DE2 manual. The signals going to the hex displays are defined in the DE2_115.qsf file. Create a VHDL entity that translates the binary values into the appropriate signals to drive the hex display. For this part, create an entity for your VHDL model that is meaningful and then import this into the entity representing the FPGA.

Include in your lab report the utilization of the various FPGA resources.

Part III

Repeat Part II, except the function is selectable using KEY[2 downto 0]:

- 000: addition
- 001: subtraction
- 010: exclusive-OR
- 011: AND
- 100: OR

- 101: multiplication (create a combinational multiplier)
- 110: division (create a combinational divider). Division should detect a divide by zero and blank the display.

As with parts I & II, include in your lab report the utilization of the various FPGA resources.

Other Requirements

Include in your lab report a comparison of the FPGA utilization for Parts I, II, and III.