VHDL Assignment #6: Getting Started With Altera DE1-SoC Board

In this VHDL assignment, you will test the one-digit binary coded decimal (BCD) adder that you designed in VHDL Assignment #5 on the Altera DE1-SoC board using the board's 7-Segment LED display as output and sliding switches as inputs.

1 Learning Outcomes

After completing this lab you should know how to

- Test digital circuits on the Altera DE1-SoC board
- Use the sliding switches on the Altera DE1-SoC board to specify the inputs to your circuits
- Use the 7-segment LED display on the Altera DE1-SoC board to display the output of your circuits

2 Prerequisites

Before starting this lab you should

- Read the sections DE1-SoC user manual as specified in the description of the assignment
- Be familiar with the information in previous coding assignments

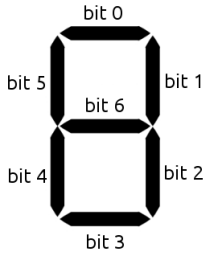
If you need any help regarding the lab materials, you can

- Ask for help from the TA of your lab session
- Refer to the text book. In case you are not aware, Appendix A "VHDL Reference" provides detailed information on VHDL
- You can also refer to the tutorial on Quartus and ModelSim provided by Intel (click here for Quartus and here for ModelSim).

It is highly recommended that you first try to resolve any issue by yourself (refer to the textbook and/or the multitude of VHDL resources on the Internet). Syntax errors, especially, can be quickly resolved by reading the error message to see exactly where the error occurred and checking the VHDL Reference or examples in the textbook for the correct syntax.

3 BCD to 7-Segment LED Decoder

A 7-segment LED display includes 7 individual LED segments, as shown below. By turning on different segments together, we can display characters or numbers. There are six of these displays on the DE1-SoC board, which you will use later to display the result of your full implementation of the adder.



We will need a circuit to drive the 7-segment LEDs on the DE1 board, called 7-segment decoder. It takes as input a 4-bit BCD-formatted code representing the 10 digits between 0 and 9, and generates the appropriate 7-segment display associated with the input code. For many LED displays, including the ones on the DE1 board, segments turn on when their segment inputs are driven low, that is "0" means on and "1" means off. This scheme for the inputs is called "active low." The VHDL code for the 7 segment decoder is provided below.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity 7_segment_decoder is
  port (  code            : in std_logic_vector(3 downto 0);
          segments_out    : out std_logic_vector(6 downto 0));
  end 7_segment_decoder;

architecture decoder of 7_segment_decoder is
begin
WITH code SELECT
segments_out <=
  "1000000" WHEN "0000", -- 0
  "1111001" WHEN "0001", -- 1
  "0100100" WHEN "0010", -- 2
  "0110000" WHEN "0011", -- 3
  "0011001" WHEN "0100", -- 4
  "0010010" WHEN "0101", -- 5
  "0000010" WHEN "0110", -- 6
  "1111000" WHEN "0111", -- 7
  "0000000" WHEN "1000", -- 8
  "0010000" WHEN "1001", -- 9
  "1111111" WHEN others;
end decoder;
```

In the next section, you will use the 7-segment LED decoder to display the inputs/outputs of the one-digit BCD adder.

4 Wrapper Design

In this part of the lab, you will design a circuit (that is called "wrapper" circuit) that performs addition of two 4-bit BCD-formatted inputs A and B, and displays the inputs as well as the result of the addition in BCD format using the 7-segment LEDs on the DE1-SoC board. You will need, therefore, to use four 7-segment LEDs on the board: the first two 7-segment LEDs will be used to display the inputs A and B, while the other two 7-segment LEDs will be used to display the result of the addition. We need two LED displays for the output as the result may be two BCD digits long. Note that you should use the binary-to-7-segment LED decoder to obtain appropriate 7-segment display codes.

Use the following entity declaration to write a VHDL description of the wrapper circuit. Note that you will need four instances of the `7_segment_decoder` component and one instance of `firstname_lastname_bcd_adder` component in your VHDL description. You can use the BCD adder that one of the members of your group designed in the previous assignment.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity firstname_lastname_wrapper is
    Port (  A, B            : in  std_logic_vector(3 downto 0);
            decoded_A       : out std_logic_vector(6 downto 0);
            decoded_B       : out std_logic_vector(6 downto 0);
            decoded_AplusB  : out std_logic_vector(13 downto 0));
end firstname_lastname_wrapper;
```

where `firstname_lastname` in the name of the entity is the name of one of the students in your group.

The wrapper circuit has two 4-bit inputs, A and B, each representing a BCD digit. The outputs are: a 7-bit display code for A, a 7-bit display code for B, and two 7-bit display codes for the sum A+B.

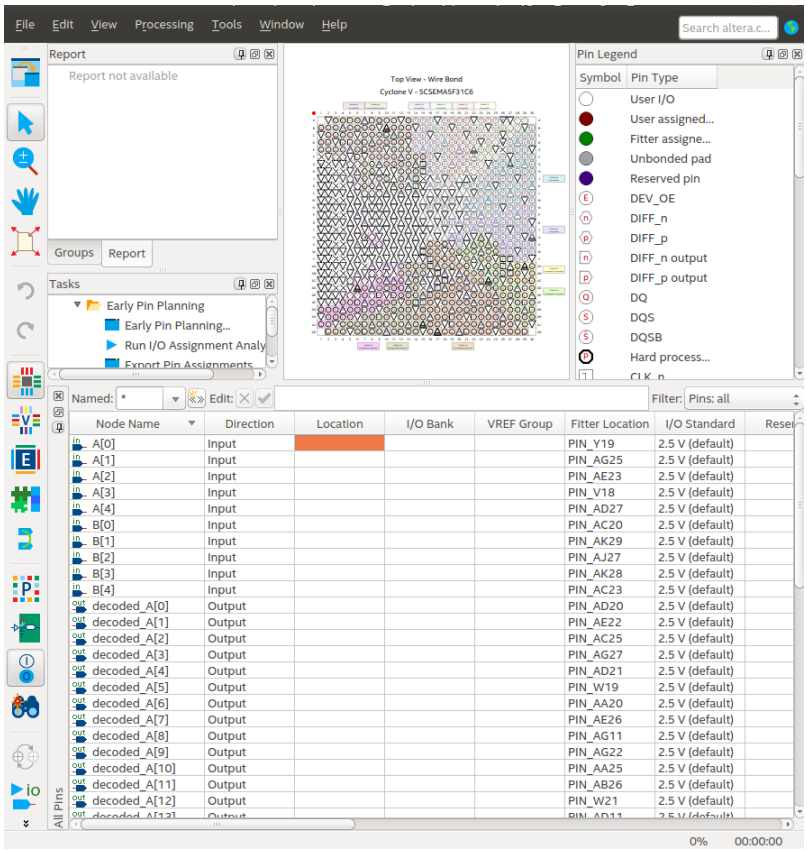
5 Testing the Wrapper on the Altera Board

You will now test the wrapper circuit you designed in Section 4. Follow the instructions described in Assignment #1 to create a project. Make sure to select the **Cyclone V** family of FPGAs, with the following part number: **5CSEMA5F31C6** when creating a project. Once created, import the VHDL description of the wrapper circuit and its components into the project and compile to make sure there are no syntax errors in your design. Once you have compiled the wrapper circuit, it is time to map it onto the target hardware, in this case the Cyclone V chip on the Altera DE1-SoC board.

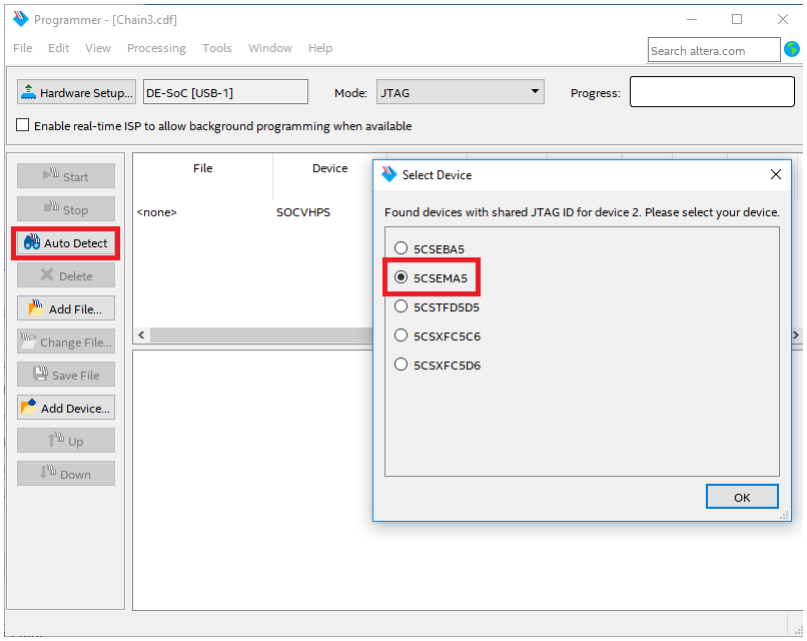
Since you will now be working with an actual device, you have to consider to which device package pins (physical pins) the various inputs and outputs of the project are connected. The FPGA chip on the board communicates with the outside world by sending and receiving binary signals using physical pins. You can think of these pins as the ports of an VHDL entity. Some of the input pins are connected to the sliding switches on the board, and some of the output pins are connected to the LED displays. In particular, you will want to connect the LED segment outputs from the instances of the `7_segment_decoder` circuit (*i.e.*, the outputs of the wrapper circuit) to the corresponding pins of the four 7-segment LED displays on the board. See Section 3.6.2 of the DE1-SoC User Manual. The mapping segments of the each 7-segment LED on the board to the pins on the Cyclone FPGA device, is listed in Table 3-9 on page 24 of the DE1-SoC User Manual.

You will also want to connect, for testing purposes, four of the slide switches on the DE1-SoC board to the input A, and another four switches to the input B of the wrapper circuit. See Section 3.6.1 of the DE1-SoC User Manual. The mapping of the slide switches to the FPGA pins is listed in Table 3-6 on page 23 of the DE1 user manual.

You must now notify the compiler to which pins the input and output signals of your wrapper circuit should be connected. For example, if B[0] is connected to switch SW0 (See Fig. 3-15 of the manual), you should notify the compiler that B[0] is assigned to pin PIN_AB12. You specify the pin assignments for your inputs and outputs by using the **Pin Planner**, which can be done by choosing the Pins item in the **Assignments** menu. See example below. Note that this figure is just an example, and does not correspond to the wrapper circuit that you are supposed to design in this assignment.



Once you have assigned all of the inputs and outputs of your circuit to appropriate device pins, re-compile your design. Your design is now ready to be downloaded to the target hardware. You may want to go over Section 4.1 of the DE1-SoC user manual for information on configuring (programming) the Cyclone V FPGA on the board. You will be using the JTAG mode to configure the device. Take the board out of the kit box, connect the USB cable to the USB port of the computer and to the USB connector on the board. Next, select the Programmer item from the Tools menu. Click Auto Detect and then select the correct device (5CSEMA5), as shown below. Both FPGA device and HPS should be detected.



Next, double-click the FPGA device (5CSEMA5), from the window that pops up, add the `.sof` file created by Quartus. Finally, check the "Program/configure" box beside the 5CSEMA5 device, and then click "Start". See also Section "Configuring the FPGA in JTAG Mode" on p. 12 of the manual. Now, you should be able to use the slide switches to insert values for inputs A and B. The 7-segment LEDs should also display inputs and outputs in BCD format.

6 Deliverables and Grading

Read this section extra carefully!

You are required to submit the following files through myCourses. Only one of the students in your group should perform the online submission, if more than one student submits files, any additional submissions will be disregarded. Note, not necessarily all Quartus-related files are to be generated in each assignment, therefore, only the relevant files for each assignment are to be submitted.

- A report that includes the answers to the questions in Section 7. Use the report template (provided in .docx format) that is attached to the assignment. Make sure that all figures are clearly visible and contain relevant information (*e.g.*, axes and relevant signal names). Save your report in a .pdf format (include the name of one of the group members in the filename).
- All design files (.vhd)
- All schematic files (.bdf)
- All testbench files (.vht)
- All design constraints files (.sdc)
- All SRAM object files (.sof)

Per the VHDL assignments submission policy, please note

- For partially submitted assignments, *i.e.*, some of the files (design/simulation/report) are missing, the penalty is 25% of the full mark of the assignment.
- For assignments where all the design/simulation files are missing, *i.e.*, only a report was submitted, the penalty is 50% of the full mark of the assignment.
- For not submitted assignments, *i.e.*, all files are missing, the penalty is 100% of the full mark of the assignment.

7 Questions

1. Provide the VHDL code that you wrote for the wrapper circuit. (1 mark)
2. Show a representative photo of the board displaying the result of the addition of A and B, where A is the last (rightmost) digit of the McGill ID number of one of the students in your group and B is the last (rightmost) digit of the McGill ID number of the other student in the group. Clearly indicate which 7-segment LEDs/sliding switches are assigned to which inputs/outputs of the circuit on the photo. (2 marks)