**Computer science department, Langara college**

**Digital Systems Design (with FPGAs)**

**Spring 2019**

# Lab 2: Combinational circuits 2

by S. Arash Sheikholeslam

In this lab, you will learn about block level implementation of combinational circuits. You will create a BCD to 7-segment converter. You will then put various combinational blocks together and make larger circuits that will manipulate the 7-segment output via the input switches. You will also learn the basics of simulation with Qsim. **Save and** **keep all the codes you produce because you will need them in the future labs.**

**Preparation:**

The following must be done before you come into the lab:

1. Read this handout before you start the lab.
2. Use only the pin assignment file provided by on D2L. Do not assign the pins manually.
3. Make sure you follow this manual’s instructions in terms on Vhdl file separation and naming.

Part 1 [3 points].:

Write a behavioral code for an entity that gets a 4-bit BCD input between 0-9 and creates a 7-segment output (7 bits.) We want the seven segment to show the letter **E** if the BCD input is larger than 9. Your entity must be named **ssg** and your vhdl file also must be named **ssg**.

Open another vhdl file and name it **Lab2A** and make a new entity (also Lab2A.) Instantiate your 7 segment as a component here and connect the BCD bits to swtiches **sw3:0**.

Simulate your design with Qsim. And observe the various 7-segment outputs for all of the possible sw4:0 inputs.

Code, compile and download your design onto the board. Show your code, simulation and working circuit to the instructor/TA and get marked for this part.

Question a) Is 7-segment active high or active low?

Part 2 [7 points]:

For this part you should write your code in another project/file names Lab2B. The goal for this part is to get a 6-bit input via **sw5:0**. You use **sw3:0** to get a 4-bit input and you produce all possible left-shifts of your input including the input itself (for instance if your input is 0001 then all of the possible left shits are 0001.

, 0010, 0100, 1000)  
Using switches **sw5:4** your circuit decides which one of the four left shifts will be displayed on the **LEDs** and show them on the LEDs.  
  
Apart from the above, you will use the **ssg** entity (without changing the entity itself) from the above to show the 4-bit output as a number on your 7-segments. However, there is a twist. We know that **ssg** will only display **BCD** numbers between 0-9 and, we know that our output is potentially larger than 9. Therefore, you need to use two 7-segments and a circuit that converts your output to a pair of **BCD**s that can be displayed on 7-segments as a double-digit number. For instance, if your LEDs show the number 0010 then we expect to see 02 on the 7-segments and if the number on the LEDs is 1101 then we should see 13.

**Performance in the lab: (10 marks)**

In this lab, you will download the circuit you designed in the Preparation on the FPGA board. The input and output pins of this FPGA are tied to the various lights and switches.

1. Read the manual carefully

1. BE SURE that you set the pin assignments before compiling your design (not doing so could damage the board!). If you have any questions about how to do this, please talk to the TA.

You must demonstrate that your circuit works to the TA or to me by the end of your lab section.

*Marking:*

Your mark for the performance part of the lab will be:

0/10: If you don’t even show up, or if you show up and don’t do anything

3/10: If you make an attempt, but really don’t get anywhere near it working

7/10: If you almost get it working, or if you get it working but can’t answer TA’s questions.

10/10: If you successfully demonstrate your design to the TA, and can answer TA’s questions.

Anything in between 0 to 3, 3 to 7 and, 7 to 10 is at instructor/TA’s discretion based on how satisfied they are with your performance and knowledge.

Note that this lab is quite easy, and you will likely finish early. Use some of the extra time to play with the software to understand what else it can do.