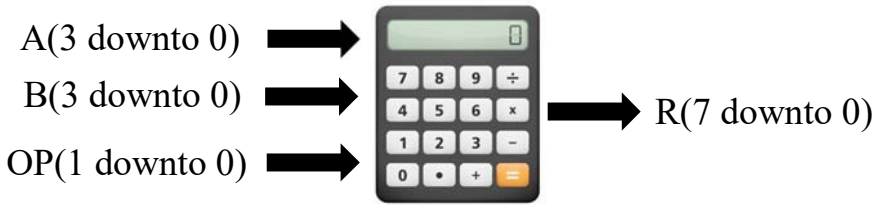
Technical Objective:

A calculator is a digital circuit that performs arithmetic operations. VHDL simplifies the design of calculator circuits by including library functions for many basic mathematical operators. When one of these functions is used in a VHDL code, the synthesizer implements the appropriate hardware. This lab investigates the design of a four function calculator in VHDL.

Pre-Laboratory: (30%)

The block diagram shown below represents a calculator. The **calculator** design has three inputs and one output. **A** and **B** are each 4-bit numbers that will be operated on according to the table below. **OP** is a 2-bit input that determines which operation is to be performed on **A** and **B**. **R** is the 8-bit result of the operation.



OP	Operation
00	A + B
01	A - B
10	A * B
11	A / B

1. Write the VHDL module (entity and architecture) for the calculator. Use a case statement.
   - Include the following libraries:
     use IEEE.STD_LOGIC_1164.ALL;
     use IEEE.NUMERIC_STD.ALL;
   - Use std_logic_vectors in the entity
   - Use signed numbers in the architecture
   - Refer to section 5.7 in the textbook for recommendations
2. Compile the VHDL
3. Submit to the dropbox prior to lab:
   - Your VHDL code

Procedure: (70%)

1. Download the calculator_tb.vhd file from MyCourses. Open the file and edit such that the component in the testbench matches exactly with your entity. If your port names are different than the ones on the component in the testbench, you will have to edit the port map as well.
2. Simulate your calculator using the testbench provided. This is a self-checking testbench that will give you an error message in the transcript window if your results are incorrect. Check the transcript window and verify there are no errors.
3. Verify the operation of your calculator by inspecting the output waveform. To make verification easier, you can change the radix for A, B and R. Choose decimal for this simulation.
4. Print and annotate comments onto the Modelsim waveform printouts. You will need 4 printouts; one example of each operation. Make sure that it is sufficiently zoomed in so that a reader can determine the values of the signals. Also make sure that your examples are significant. (i.e., don't demonstrate correct multiplication functionality by multiplying with 0).
5. Obtain a signoff when you are sure your simulation is correct.
6. In Quartus, assign pins for the DE2 board. The inputs should be on switches and the ouputs on LEDs. Recompile the project with the pin assignments. Program the board.
7. Verify that your board is working and obtain a signoff.

Documentation:

There is no documentation due for this lab. Save the signoff sheets and annotated waveforms for lab 7.

Signoffs and Grade:

Please submit with Lab 7 Report

Name:_____

Component	Signoff	Date	Time
Functional Simulation (35 pts)			
Working Board (35 pts)			

==========

Component	Received	Possible
Prelab		30
Signoff		70
Penalties • after the first 15 minutes of lab session 6: -10 • after the first 15 minutes of lab session 7: -25 • no signoffs after the first 15 minutes of lab session 8:	-	
Total		100