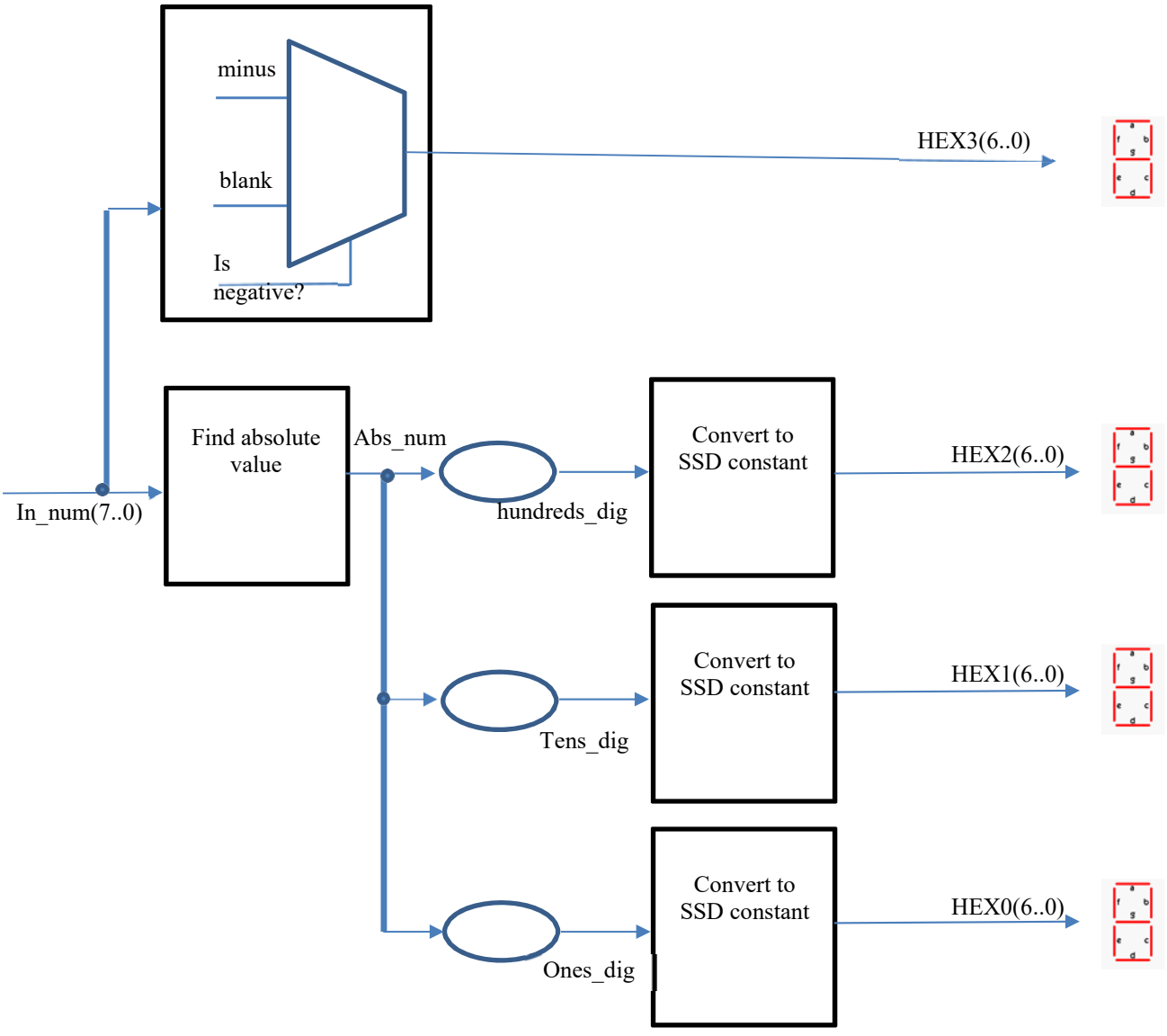
Technical Objective:

Taking advantage of the concurrent nature of VHDL, large systems can be broken down into functional blocks. Data passes between functional blocks through the use of signals. In this lab we will investigate the design of a digital system where each functional block is written as a separate process. The system to be designed has an 8 bit STD_LOGIC_VECTOR input and outputs the constants to display the input's signed equivalence on four 7-segment displays.

Pre-Laboratory: (30%)

Consider the system diagram below. Each rectangle represents a functional block and thus should be its own process. The 8-bit input is read in two processes. The first process finds the absolute value and the second puts a negative sign or a blank on the leftmost Hex display. The absolute value of **in_num** is separated into its three digits using division (/) and remainder (rem) functions. Each of the digits is input to a process that outputs the appropriate constant to display that digit on a seven segment display.



1. Write the VHDL for the binary to seven-segment-display system shown above. Your VHDL must have at least five separate processes, each having only one output.
2. Submit your code to the dropbox in MyCourses prior to your lab session.

Procedure: (70%)

1. Download the Display_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 binary to seven-segment-display system module using the testbench provided.
3. Download the wave.do file from MyCourses and store it with the rest of your files. In the transcript window in Modelsim type “do wave.do” and then enter. In the wave window, select each of the hex outputs and right click. Choose radix > radix_ssd. This is a custom radix that will convert your constants to display characters.
4. Verify the operation of your module by inspecting the output waveform. The ssd radix selected in step 3 should show in yellow the character that will be displayed on the seven segment display.
5. Print and annotate comments onto the Modelsim waveform printouts.
6. Obtain a signoff when you are sure your simulation is correct.
7. In Quartus, assign pins for the DE2 board. The 8-bit input should go on SW7-SW0.
8. Recompile the project with the pin assignments. Program the board.
9. Verify that your board is working and get a signoff.

Documentation:

There is no documentation due for this lab. Save you VHDL code, annotated waveforms and signoffs for your lab 7 report.

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 7: -10 • after the first 15 minutes of lab session 8: -25 • no signoffs after the first 15 minutes of lab session 9:	-	
Total		100