Laboratory 5: Hardware Add and Subtract [8 bit] with State Machine

1 Introduction

In the last lab you created a basic 3 bit add/sub circuit that used the DE1 SoC switches for inputs A and B. Due to the fact that the system did not keep track of any 'state', both A and B inputs had to be selected at the same time, thus limiting the size of the add/sub circuit. In this lab we will make use of the concept of 'state' and enter values of A and B one at a time. You will be creating an 8 bit add/sub circuit and displaying the result on the seven segment display.

1. On reset, shall enter the 'input_a' state and input 8 bits for the first input of the add/sub circuit via the slide switches and display the value on 3 seven segment displays.
2. Upon pressing a button, shall enter the 'input_b' state and input 8 bits for the first input of the add/sub circuit via the slide switches and display the value on 3 seven segment displays.
3. Upon pressing a button again, shall enter the 'disp_sum' state and the sum of A and B shall be displayed on 3 seven segment displays.
4. Upon pressing a button again, shall enter the 'disp_diff' state and the difference of A and B shall be displayed on 3 seven segment displays.
5. Upon pressing a button again, shall enter the 'input_a' state and operate as described above.
6. Shall have an active high reset switch.
7. Shall operate with unsigned base 512 numbers. Ex. 0 - 1 = 511
8. All inputs shall be synchronized with the 50 MHz clock.
9. All 3 seven segment displays shall display a maximum decimal number of 999. [Although 512 is max realistically].
10. Shall display the present state via 4 LEDs.

2 Pre-Lab [Block Diagram]

You shall create a block diagram with a state transition diagram accurately mapping out your design which is due 1 hour before lab in the myCourses dropbox. Failing to submit an electronic version to dropbox 1 hour before the lab will result in a zero for the prelab. Also bring a printout of the block diagram to your lab class for an open discussion. Make sure to include proper synchronization, edge detection, and signal names in your diagram as well as a clear notation indicating the bit width of various busses. A state transition diagram is also required.

3 SIMULATION

Create a simulation that tests out the below use cases. You do not need to use assertions for this lab. I have posted my double_dabble.vhd file which is a process that converts a 12 bit binary number to 3 seven segment display nibbles. You will need this code to adequately display the std_logic_vectors via the 3 seven segment displays.

A	B	SUM	DIFF
5	2	7	3
2	5	7	509
200	100	300	100
100	200	300	412

4 HARDWARE

Target your design onto the DE1 SoC and receive a signoff. To save you time, I posted my compile.tcl file that lists the pin assignments that I used.

5 DELIVERABLES

To receive full credit for this lab one must hand in the below items no later than 168 hrs [7 days] after the start of one's lab session. Signoffs can be obtained after the due date as long as the time stamp of the code is from before the deadline.

- ☐ Hard copy of this document.
- ☐ Hard copy of all src files [no tabs and print from notepad++ with 'show symbols' on].

6 SIGNOFFS

Category	Initials	Date	Points
Block Diagram			/20
Simulation			/30
Demonstration			/40
Deliverable			/10
Final Grade			/100