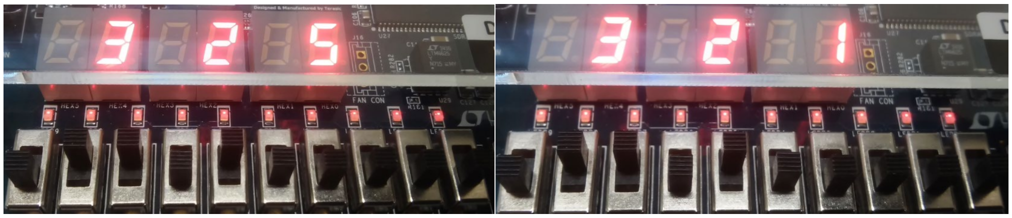
Laboratory 4: Hardware Add and Subtract [3 bit]

1 Introduction

In the past couple of labs we have been focusing on scripting, folder structure, and general VHDL design principles. In this lab you will have the chance to combine several sub-components in order to create a basic add/subtract circuit on the DE1 SoC. Your design:

1. Shall input 3 bits for the first input of the add/sub circuit via the slide switches.
2. Shall input 3 bits for the second input of the add/sub circuit via the slide switches.
3. Shall have an active high reset switch.
4. Shall display both the first and second inputs on seven segment displays.
5. Shall display the result of an addition on a seven segment display when the add button is pressed.
6. Shall display the result of a subtraction on a seven segment display when the subtract button is pressed.
7. Shall operate with unsigned base 16 numbers. Ex. 0 + 2 = 2, 1 – 2 = 15, 0 – 7 = 9
8. All inputs shall be synchronized with the 50 MHz clock.
9. May display numbers greater than 9 on the result LCD by using a,b,c,d,e,f to represent Hex but this is optional.



Sample Addition and Subtraction Values.

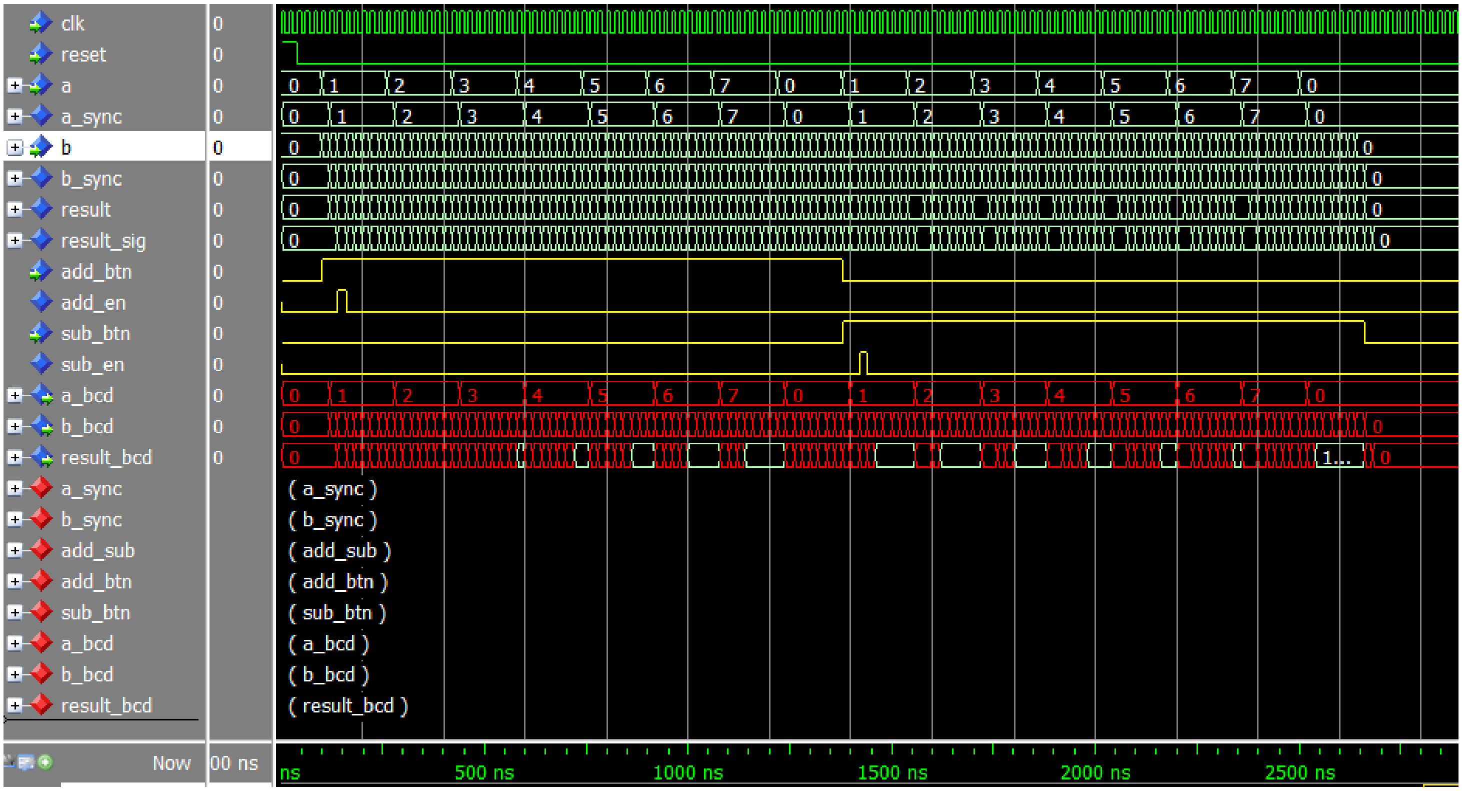
2 Pre-Lab [Block Diagram]

<u>You shall create a block 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.</u> 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. **Hint: the inputs are 3 bits however the output of the add/sub block is 4 bits. Make sure to make use of proper guard bits.** Think through such questions as:

- Will I have an add_sub.vhd file or an add.vhd and sub.vhd file?
- Will I have a flag sent to my add_sub.vhd file to select whether to do an addition or subtraction, or will I return both an add and sub signal and make the selection in the top module.
- How many files will I have and what will they be named?
- What modules are synchronous and which one are asynchronous?

3 SIMULATION

You can use your own signal names however you must cycle through all possible inputs. Make sure to simulate an add push button as well as a subtract push button as shown below. Also be sure to include the red BCD radix from the previous lab as shown below. Once you are confident that your simulation is working, receive a signoff. Note that assertions are not required for this test bench. Color coding your signals as shown in the diagram below will help clarify your design for the signoff.



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