CS223 Laboratory Assignment 5

Serial Adder

Preliminary Work Submission Deadline:
All Sections: 22nd May, Mon 12:00

Lab dates and times:
Section 1: 25th May, Thu 13:30-17:20 in EA-Z04
Section 2: 22nd May, Mon 13:30-17:20 in EA-Z04

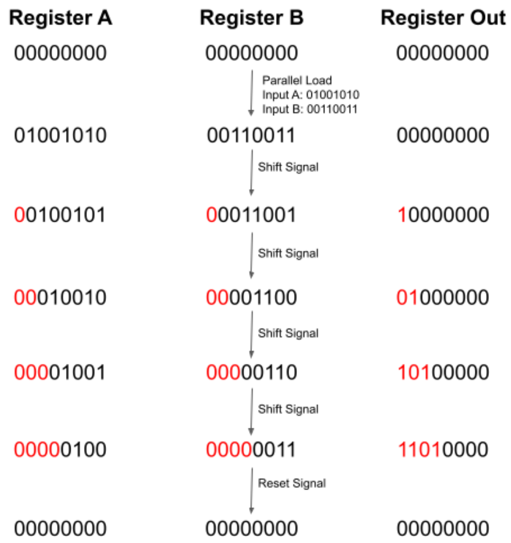
Location: EA Z04 (in the EA building, straight ahead past the elevators)
Groups: Each student will do the lab individually. Group size = 1

Lab Assignment

In this lab, you are going to design an unsigned 8-bit serial adder using synchronously resettable D flip-flops. Your serial adder should consists of three 8-bit shift registers (two of them are for the addends A and B, and the other one is for the sum result), one 1-bit full adder, and one separate D flip-flip for storing the carry variable. Your serial adder should be controlled by three signals:

- **Shift Signal:** When the shift signal is received, one shift right operation should be performed **for all shift registers**. It basically means performing the addition operation of the next bits. Note that no shifting operation should be performed after all the summation is completed.
- **Parallel Load Signal:** When the parallel load signal is received, two 8-bit external data inputs (one for each A and B addends) should be loaded to the corresponding registers.
- **Reset Signal:** When the reset signal is received, the outputs of all D flip-flops should be cleared.

An example workflow is as follows:



Register A	Register B	Register Out
00000000	00000000	00000000
	Parallel Load Input A: 01001010 Input B: 00110011	
01001010	00110011	00000000
	Shift Signal	
00100101	00011001	10000000
	Shift Signal	
00010010	00001100	01000000
	Shift Signal	
00001001	00000110	10100000
	Shift Signal	
00000100	00000011	11010000
	Reset Signal	
00000000	00000000	00000000

Preliminary Work (30 pts)

The content of your preliminary work report will be as follows:

(a) A cover page including course code, course name, and section, the number of the lab, your name-surname, student ID, date.

(b) Circuit schematic for your shift register design using D flip-flops.

(c) Circuit schematic for your serial adder using the shift registers, full adder, and D flip-flop.

(d) SystemVerilog module for synchronously resettable D flip-flop.

(e) Structural SystemVerilog module for your shift register using the D flip-flop module along with the testbench.

(f) Structural SystemVerilog module for your serial adder using the shift register, full adder, and D flip-flop modules along with the testbench.

Additional pre-lab work:

You should read the following documents (available on Moodle) to be familiar with steps of design flow (Simulation, Synthesis, Implementation, Bitstream Generation, Downloading to FPGA board), using Xilinx **Vivado** tool. You can download, install and practice working with Xilinx Vivado on your own computer with a free webpack license.

- Suggestions for Lab Success.
- Basys 3 Vivado Decoder Tutorial.
- Vivado Tutorial.
- Basys 3 FPGA Board Reference Manual.

Simulation (20 pts)

Using the SystemVerilog module for the serial adder and testbench code you wrote in the preliminary part (f), verify in simulation that your circuit works correctly and show the results to your TA.

Implementation on FPGA (50 pts)

In this step, you will implement your modules on the FPGA board. The signals should be controlled via the pushbuttons on the Basys 3 board. The left pushbutton should correspond to the reset signal, the middle pushbutton should correspond to the shift signal, and the right pushbutton should correspond to the parallel load signal (Figure 1). For parallel loading, the **unsigned** external data inputs should be determined by the switches on the Basys 3 board (Figure 2). The left-most 8 switches should correspond to the input A and the right-most 8 switches should correspond to the input B. Finally, the value stored in the output register should be displayed on the right-most 8 LEDs. Note that you need to pay attention to the order of the bits regarding the 8-bit value representation. When you are convinced that your design works correctly, show it to your TA.

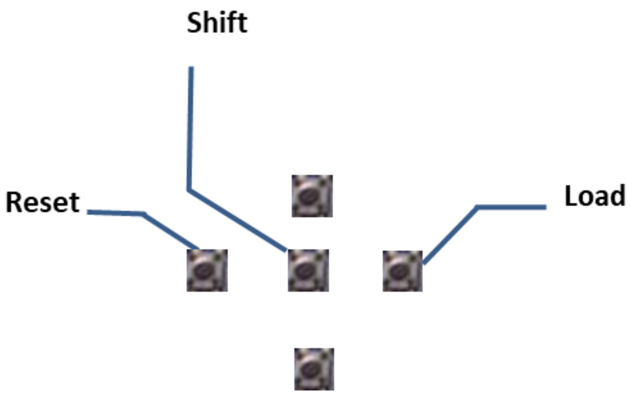


Figure 1: Reset, Shift, and Parallel Load Buttons

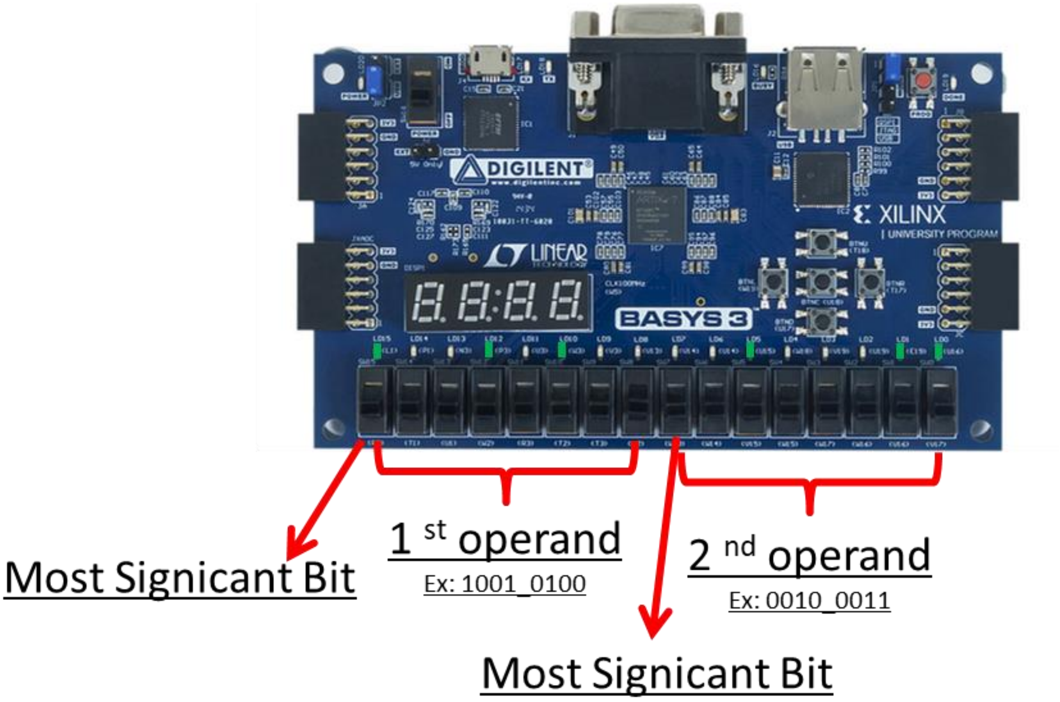


Figure 2: External Data Inputs

Submit your code for MOSS similarity testing

Finally, when you are done and before leaving the lab, you need to upload the file *StudentID_SectionNumber.txt* created in the Implementation on FPGA part. Be sure that the file contains exactly and only the codes which are specifically detailed above. If you have multiple files, just copy and paste them in order, one after another inside text file. Check the specifications! Even if you didn't finish or didn't get the SystemVerilog part working, you must submit your code to the Moodle Assignment for similarity checking. Your codes will be compared against all the other codes in all sections of the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that the code you submit is the code that you actually wrote yourself! All students must upload their code to the 'Moodle Assignment' specific for their sections. Check submission time and don't miss it before leaving the lab.