CS223 Laboratory Assignment 5

Reduce Sum on Array

Lab dates and times:

Section 1:	05.12.2020 Saturday	08:30-11:30
Section 2:	05.12.2020 Saturday	15:30-18:30
Section 3:	06.12.2020 Sunday	15:30-18:30
Section 4:	04.12.2020 Friday	08:30-11:30
Section 5:	07.12.2020 Monday	13:30-16:30
Section 6:	08.12.2020 Tuesday	13:30-16:30

This lab will be performed online through Zoom.

This lab, you are asked to implement a HLSM that will sum all the elements of an array. The array should be implemented as a RAM which can hold at most 16 elements, where each element is 8-bits. The RAM you implement must have the following inputs: clock, writeAddress, writeData, writeEnable, readAddress1, readAddress2, readData1, readData2. You will use switches to enter data, where the SW[7:0] (8 leftmost switches) would define the data to enter and SW[15:12] would define the address in memory. When the uppermost pushbutton is pressed, data defined by switches will be written to the specified address of the memory. Left and right pushbuttons will be used to circulate and show the specified memory address and data on seven-segment display. Middle pushbutton will calculate the sum and show it on LEDs. You will be given the seven-segment display and debouncer for pushbutton modules. One of the read ports of the RAM will be connected to seven-segment display to show the data, and the other one will be connected to reduce sum module which will calculate the sum of all elements. In the Seven-Segment Display, the most significant digit will show the current address to display, and the least significant 2 digits will show the data in that address. The remaining digit in between should be turned off.

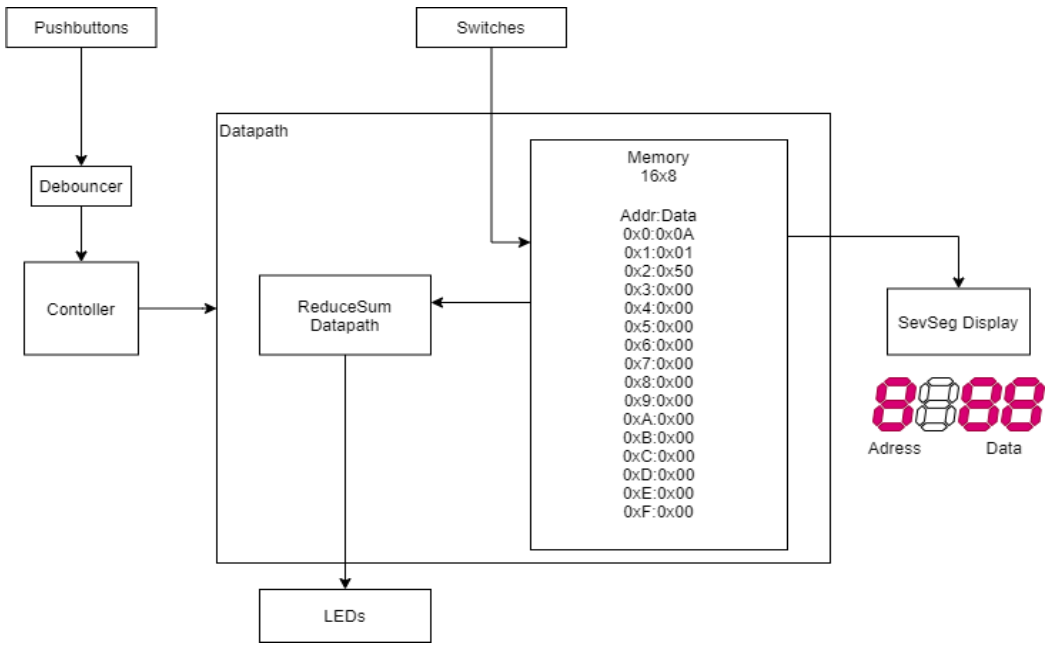

Figure 1: Flow Architecture

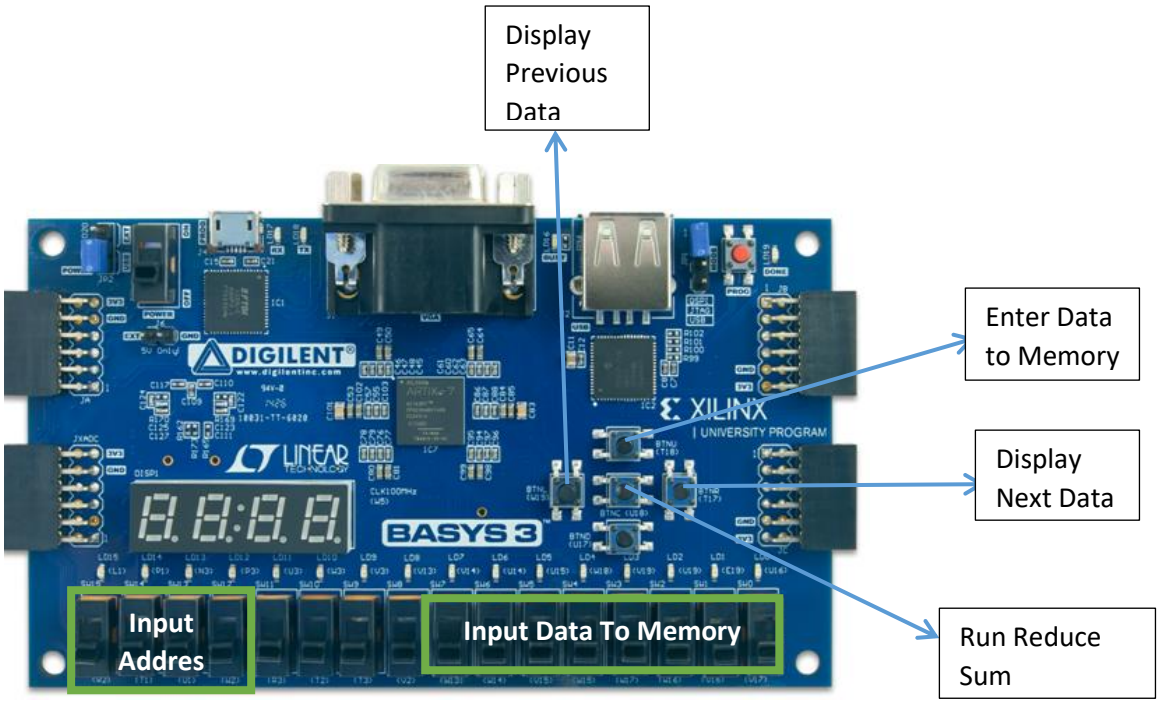

Figure 2: Data Interface Inputs

For the example memory given in Figure 1, if the memory address 0 is displayed, seven segment will show “0 0A,” if then display previous data button is pressed, “F 00” should be displayed. If Run Reduce Sum button is pressed, the leds will show 0x5B in binary as the sum. Assuming input address switches are “1111” and input data to memory switches are also all on, when display enter data to memory button is pressed, the seven segment should now show “F FF” as adress 0xF was being displayed and the data is updated to “0xFF” now. LEDs will not change until Run Reduce Sum button is pressed again.

Preliminary Report (45 points)

Today's lab needs considerable advance preparation. These advance designs and SystemVerilog models should be prepared in advance, and assembled neatly into a Preliminary Report with a cover page and pages for the SystemVerilog codes. Each page should have a proper heading. The report should be uploaded on Moodle as a pdf file before the start of the lab. The content of the report will be as follows:

a) Show the ReduceSum HLSM. Show also the controller/datapath diagram for it.
b) You will be given a debouncer module for your pushbuttons. Research and explain briefly why there is a need for such a circuit.
c) Write the SystemVerilog Code for your memory module. Write a testbench for it.
d) Write the SystemVerilog Code for your ReduceSum Module. Write a testbench for it.
e) Write the SystemVerilog Code for the top design which will also include the Seven Segment Display and debouncer modules.

Simulation (20 Points)

Enter System Verilog module for your ReduceSum module and run your testbench for it. Show your simulation to TA.

Implementation on FPGA (35 Points)

In this part you are going to implement your overall module on FPGA and have a demo.

1) Right and left pushbuttons will be used to circulate in memory. Initial address will be 0. If right pushbutton is pressed the data on address 1 should be displayed. Then if again the right pushbutton is pressed data on address 2 should be displayed an so on. If left pushbutton is pressed while address is 0, data on address 15 should be displayed.
2) When the upper pushbutton is pressed, the data on SW[7:0] should be put in address SW[15:12]. Initally, all the values should be 0 in memory.
3) With the press on middle pushbutton, the ReduceSum should be calculated and and the result should be displayed on LEDs.

Now test your code and show the result to your TA.

Submit your code for MOSS similarity testing

Finally, when you are done and before leaving the lab, you need to copy all the SystemVerilog codes you wrote to a txt file named *StudentID_name.txt* and upload it to Moodle. If you have multiple files, just copy and paste them in order, one after another inside text file. Even if you didn't finish or didn't get the SystemVerilog part working, you must submit your code to the Moodle 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 indication of plagiarism). So be sure that the code you submit is the code that you actually wrote yourself!

Clean Up!

1) Clean up your lab station, and return all the parts, wires, the Beti trainer board, etc. Leave your lab workstation.

2) CONGRATULATIONS! You are finished with this lab and are one step closer to becoming a computer engineer.

NOTES

--Advance work on this lab, and all labs, is strongly suggested.

--Be sure to read and follow the Policies for CS223 labs, posted in Unilica.

LAB POLICIES

1. There are three computers in each row in the lab. Don't use middle computers, unless you are allowed by lab supervisor.
2. You borrow a Lab-board containing the development board, connectors, etc. in the beginning. The lab supervisor takes your signature. When you are done, return it to her, otherwise you will be responsible and lose points.

3. Each Lab-board has a number. You must always use the same trainer board pack throughout the semester.
4. You must be in the lab, working on the lab, from the time lab starts until you finish and leave. (Bathroom and snack breaks are the exception to this rule). Absence from the lab, at any time, is counted as absence from the whole lab that day.
5. No cell phone usage during lab. Tell friends not to call during the lab hours--you are busy learning how digital circuits work !.
6. Internet usage is permitted only to lab-related technical sites. No Facebook, Twitter, email, news, video games, etc--you are busy learning how digital circuits work !.
7. If you come to lab later than 20 minutes, you will loose that session completely.
8. When you are done, DO NOT return IC parts into the IC boxes, where you've taken them first. Just put them inside your lab pack box. Lab coordinator will check and return them later.