CS224 : Assignment 3

Date of Demonstration 21st February 2023, Lab Timing (2PM-5PM), 14% (8%+ 8%) weight

Part I (Breadboard and IC part)

Design and Implement an 8-bit unsigned multiplier (using ICs and breadboard). Assume both the inputs are unsigned 8 bits integers and produce a 16-bit unsigned integer output.

Some hints:

- Use two 4-bit Adders/ALUs to do addition operations
- Use DIP switches for inputs
- Use a combination of both SIPO and PISO to make full-fledged universal shift register (8-bit)
- For multiplicand (A of AxB), you can use two 4-bit PIPO registers
- For the controller, you can use a mod 8/16 counter
- A combination of D-FFs can be used as a register if necessary
- Display all the outputs in 7 Seg LEDs, except the carry-out. If you are not getting a sufficient number of Seven Segment LEDs then used LEDs of Breadboard for output. Caution: don't connect external 7 Segment LED without resistors.

Ref1: "RTL design for Multiplier", Chapter 8, Section 8.7 of "Digital Design" 4th Edition Book by M. Moris Mano.
Ref2: "Multiplier Design", Chapter 3, Section 3.3 of "Computer Organization and Design" by L. Henessy and D Paterson.

You can use Booth Encoding if you use ADDer/SUBstractor.

Part II (VHDL and FPGA) : Design of Register Based toy Intruction Processor (without MemOPs and BrachOPs)

Design and implement RTL processor operations demonstrator where instruction size is 6-bit and instruction has a 3-bit operations field and 3-bit for register selection. Assume your design has a register file (RF) with eight 8-bit registers and an accumulator (ACC) register (8-bit).

Support 3 bit operations:

000	ADD R	//ACC =ACC+R
001	SUB R	//ACC=ACC-R
010	MRA R	//Move RF[R] value of ACC
011	MAR R	//Move ACC value to RF[R]
100	SCAN	//Store the SWITCH value (SW8-SW15) to ACC
101	PRINT	//Display Content of ACC to 7 SEG LED

Use input switches SW2-SW7 of FPGA board for Instruction. SW2-SW4 for operation, SW5-SW7 for register number. Use input switches SW8-SW15 of FPGA for value of register in SCAN operation. Use input switches SW0 and SW1 for control. ***When SW0 goes from low to high perform the operation.***

```
Hint : for top module
reg RI[8], SEG7[8]; //Connect RI to SW8-SW15 and SEG7 to 7 Seg port/output
reg R[8][8]; // or in bit R[7 downto 0][7 downto 0]
always @(rising edge of SW0){
   switch  (SW2-SW4){
   case 000 : ACC= ACC+R[SW5-SW7]
   case 001 : ACC= ACC-R[SW5-SW7]
   case 010 : ACC= R[SW5-SW7]
   case 011 : R[SW5-SW7]=ACC
   case 100 : ACC= RI ; // SW8-SW15
   case 101 : SEG7=ACC ; //PRINT VALUE of ACC to 7SEG
   }
}
```

Finally you need to read five eight bit numbers and Display the sum.

Your program should be like

```
Instruction      InputSwitch Meaning/Description

SCAN             100XXX      //ACC=SW8-SW15
MAR 000          011000      //R0=ACC
SCAN             100XXX      //ACC=SW8-SW15
MAR 001          011001      //R1=ACC
SCAN             100XXX      //ACC=SW8-SW15
MAR 010          011010      //R2=ACC
SCAN             100XXX      //ACC=SW8-SW15
MAR 011          011011      //R3=ACC
SCAN             100XXX      //ACC=SW8-SW15
ADD 000          000000      //ACC =ACC+R0
ADD 001          000001      //ACC =ACC+R1
ADD 010          000010      //ACC =ACC+R2
ADD 011          000011      //ACC =ACC+R3
MRA 100          010100      //R[4]=ACC
PRINT            101XXX      //PRINT=ACC
```

Evaluation Procedure

- All the member of the group need to be present at the time of Demonstration of the assignment. All the absent members will be awarded 0 marks for the assignment. Please show your ID card at the time of demonstration (as it is difficult to remember faces of all the 128 students of your batch).
- Grading will be based on (a) Correctness, (b) Quality of design, (c) Wire optimization, (d) Optimum number of chip used,(e) Cleanliness in design (Wire and Chips should be organized to look good), (f) Use of proper Comment/Naming/Labeling of the wires and (g) Questionnaire and explanation.
- For HDL codes the quality will be based on FPGA minimum resource utilization (Synthesis Report: optimized number of LUTs, register, Minimum Clock), coding style (Use of proper Comment/Naming/Labeling of the wires), performance, comments, and questionnaire and explanation.