

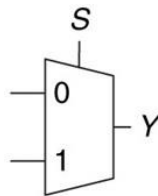
CS223 – DIGITAL DESIGN

HOMEWORK # 2

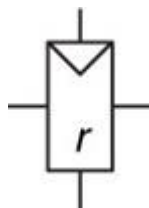
- Q1.** Given the logic function $F(w, x, y, z) = \sum m(0, 1, 2, 6, 10, 13) + \sum d(4, 8, 15)$ where m represents minterms of the function and d represents don't cares. Find the minimal (optimal) Boolean expression for F using the Karnaugh map method.
- Q2.** Find a minimal Boolean expression for the function given in the following Karnaugh map. Implement the function F using only NOR gates assuming that the complements of input variables are available.

CD \ AB	00	01	11	10
00	1		X	1
01			1	
11	X	1	X	1
10	X		X	1

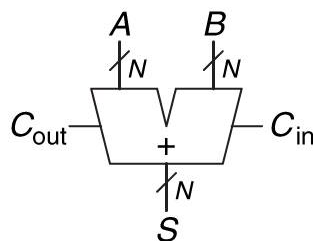
- Q3.** Design a 4-bit left shifter using only eight 2:1 multiplexers. The shifter accepts a 4-bit input $A_{3:0}$ and a 2-bit shift amount $S_{1:0}$. It produces a 4-bit output $Y_{3:0}$. Write the truth table and sketch the circuit schematic using the multiplexer symbol given below.



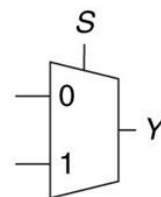
- Q4.** You will design an 8-bit synchronous Up/Down counter. The inputs are CLK, Reset, and Up. When Reset is 1, the output bits $C_{7:0}$ are all 0. Otherwise, when $Up = 1$, the circuit **counts up**, and when $Up = 0$, the circuit **counts down**. You can use only one 8-bit **register** with reset input, one 8-bit **adder**, and one 8-bit 2:1 **multiplexer**. Sketch the circuit schematic of your design using the component symbols given below.



Register



Adder



Multiplexer

Q5. For the logic function $F = BD(\bar{A} + C) + A(C + \bar{D}) + \overline{(B + \bar{C})(A + \bar{B} + C + D)}$

(a) Write the truth table.

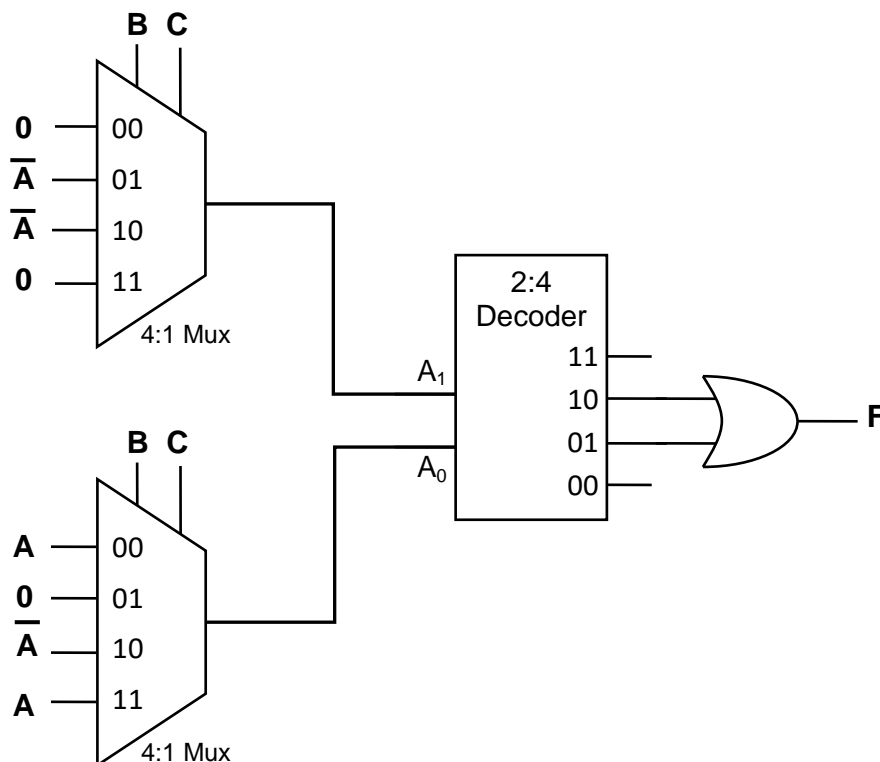
(b) Write the function in Sum-of-Products (SOP) and Product-of-Sums (POS) canonical forms.

Q6. Let us define the distance between two binary numbers as the number of bit positions that the numbers differ from each other. For example, the distance between 001 and 100 is 2 since they differ in two-bit positions. The distance between 000 and 111 is 3 since they differ in three-bit positions. The distance between 010 and 000 is 1 since they differ in one-bit position.

Design a combinational circuit that will take two 3-bit binary numbers ($A_2A_1A_0$ and $B_2B_1B_0$) as input and will produce a 2-bit binary number (Z_1Z_0) as output. The output will give the distance between input binary numbers. For example, if the inputs are $A_2A_1A_0 = 101$ and $B_2B_1B_0 = 011$, then the output will be $Z_1Z_0 = 10$ indicating that the distance is equal to 2.

Draw the logic diagram of your circuit. You can use **only** AND, OR, NOT, and XOR gates.

Q7. Find the minimized Boolean expression for the logic function performed by the following circuit (inputs are A, B, C and the output is F).

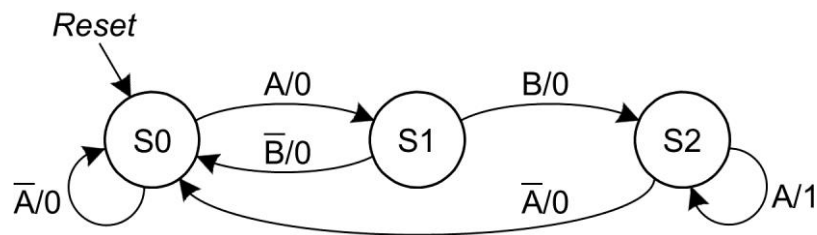


Q8. Design a circuit that will find the absolute value of a binary number represented by using two's complement. Use this circuit to calculate $|A-B|$ where A and B are 8-bit numbers in two's complement form. You are allowed to use adders, multiplexers, and logic gates. *Show your design clearly by drawing the connections and write the number of bits.*

Q9. The state transition diagram of an FSM is given below. The FSM receives two inputs **A, B** and it has one output **Y**.

(a) Write the state transition and output table using binary state encodings.

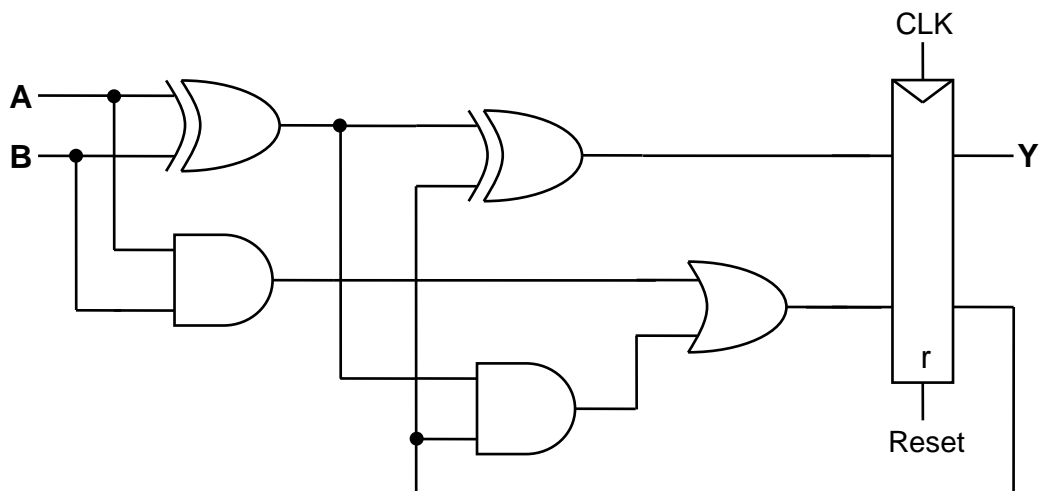
(b) Write the minimized Boolean equations for the next-state logic and output logic.



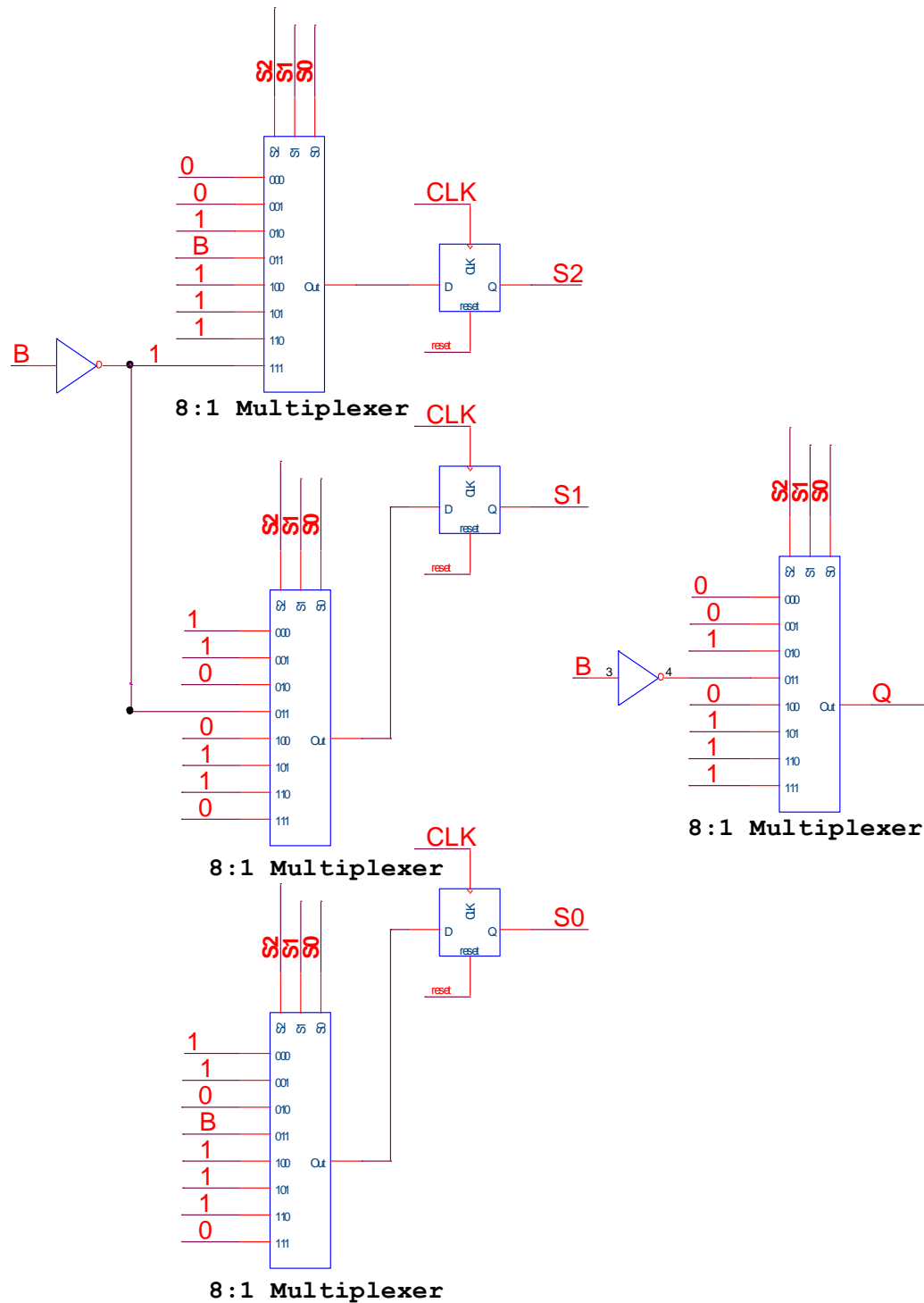
Q10. Analyze the Finite State Machine (FSM) given below.

(a) Write the state table and output table for the FSM using binary state encodings.

(b) Sketch the state transition diagram of the FSM.



Q11. For the sequential circuit (FSM) given below (**B** is the input, **Q** is the output)



(a) Write the Boolean equations for the next-state and output logic in sum-of-products form.

$$S'_2 =$$

$$S'_1 =$$

$$S'_0 =$$

$$Q =$$

(b) Write the state transition and output tables given below.

Present State			Input	Next State			Output
S ₂	S ₁	S ₀	B	S' ₂	S' ₁	S' ₀	Q
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

(c) Draw the state transition diagram for this FSM. Disregard (don't draw) any states that cannot be entered.

(d) Analyze the finite state machine and in one or two sentences, tell what it does. You should summarize the overall function of the FSM. Do not go into details like "when in this state, if such-and-so happens, then this happens...", etc.

Q12. Design a **Mealy** Finite State Machine (FSM) that receives a bit string of 1's and 0's at the input and outputs 1 whenever it detects a sequence of 010; otherwise outputs 0. Draw the state transition diagram of your FSM. Write the state transition and output table for the FSM using binary state encodings.

Q13. Design a system that counts the number of events on a single-bit input B and always outputs two unsigned numbers on 16-bit outputs C10 and C01, which are initially 0. An event is a change from 0 to 1 (C01 is incremented by 1) or from 1 to 0 (C10 is incremented by 1). Assume the system count rolls over when the maximum value of either C10 or C01 is reached.

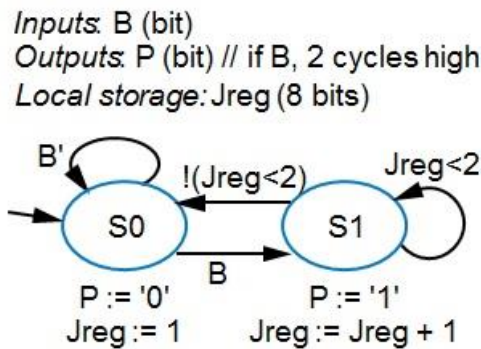
(a) Capture the system behavior as an HLSM and draw the state transition diagram.

(b) Create a datapath for your HLSM. Label the input/output signals of all components.

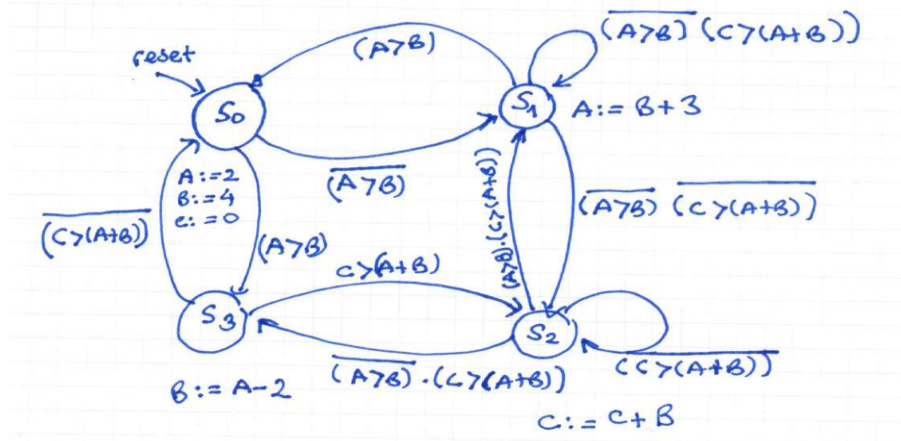
Q14. Design a high-level state machine (HLSM) that counts the number of 0's in an 8-bit input **A** and outputs the result on a 4-bit output **C**, which is initially 0. The computation starts when a single-bit input **b** changes from 0 to 1. The input **b** becomes 1 for one clock cycle and then it returns to 0. The computed result should be held at the output until the next time **b** changes from 0 to 1 and a new value is computed.

- Capture the system behavior as an HLSM and draw the state transition diagram.
- Create a datapath for your HLSM. Label the input/output signals of all components.
- Connect the datapath to the controller and show all signals between two units.
- Derive the FSM and draw the state transition diagram.

Q15. For the HLSM given below, complete the RTL design process. Create a datapath with each component clearly labeled and show the connections and number of bits. Connect the datapath to the controller and show all signals between two blocks. Finally, convert the HLSM to a finite-state machine (FSM) for the controller.



Q16. The state transition diagram for an HLSM is given below. The output **C** (32 bit) comes from a register of the same name (Creg). **A** and **B** are 8-bit local storage registers.

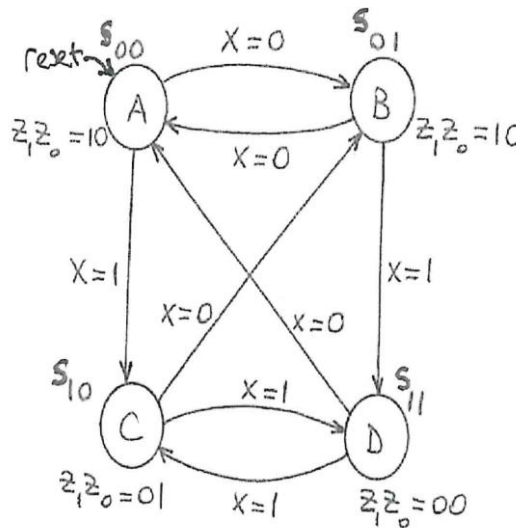


- Create a datapath for this HLSM. Show and name all the inputs and outputs of the datapath. For each component in the datapath, clearly identify it to show its function, name its inputs and outputs, and give the number of bits of each.

- (b) Derive the FSM and draw the state transition diagram. Write the state transition and output tables using binary state encodings (Hint: You can use a shorthand notation Agt for $A > B$ and Cgt for $C > A+B$).
- (c) Write the minimized Boolean equations for the next state and output of FSM.

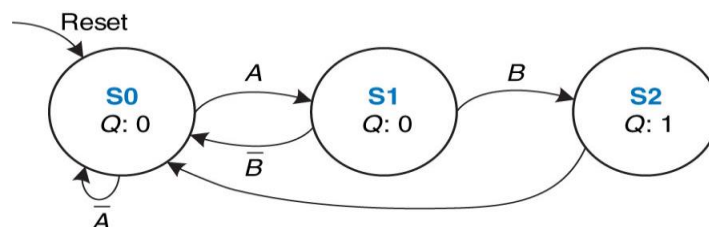
Q17. Design a circuit (HLSM) with a clock input and two outputs z and y . The circuit will generate output $zy = 00$ for 5ms then $zy = 01$ for 25ms then $zy = 10$ for 12ms and $zy = 11$ for 6ms. You are given a 1 MHz clock signal, up-down counters, multiplexers, registers, D flip-flops and logic gates. You may also use a timer in your circuit. *Show your design clearly by showing the connections and number of bits.*

Q18. The state transition diagram of an FSM is given below. You will design the controller for this FSM using only two 4:1 multiplexers and two 2:1 multiplexers.



- (a) Write the state transition table and output table for the FSM using binary state encodings.
- (b) Write the Boolean equations for the next state and output of FSM.
- (c) Sketch the circuit schematic of FSM using state register, two 4:1 multiplexers, and two 2:1 multiplexers. Don't use any logic gate. You can use the logic levels 1 and 0 as input.

Q19. Design a Finite State Machine (FSM) using the state transition diagram given below



- (a) Using binary state encodings, write the state transition table and output table for the FSM. Write Boolean equations for the next state and output logic.
- (b) Sketch a schematic for your FSM circuit.

- Q20.** A register file has 16 registers each having 16 bits. The registers RF[0] to RF[15] contain positive or negative integers. Design an HLSM which will find the sum of absolute values of the 16 integers stored in RF[0] to RF[15]. The negative numbers are in 2's complement form. The sum is going to be stored in a separate **SUM** register. Upon reset, the HLSM will be in the initial state waiting for the **GO** signal. When the operation is completed the HLSM will return to its initial state and wait for a GO signal to start over again. *Show your design clearly.*
- (a) Capture the system behavior as an HLSM and draw the state transition diagram.
 - (b) Design the datapath for this HLSM. Specify each component in the datapath clearly (i.e. name of the component, input signals, output signals, number of bits, etc.).
 - (c) Connect the datapath to the controller (FSM) and show the signals between the controller and the datapath.
 - (d) Derive the FSM and draw the state transition diagram.