King Fahd University of Petroleum and Minerals
College of Computer Science and Engineering
Computer Engineering Department

COE 202: Digital Logic Design (3-0-3)
Term 191 (Fall 2019)
Final Exam
Thursday Dec. 26, 2019

Time: 150 minutes, Total Pages: 11

Name:________________________________ **ID:**________________ **Section:** _______

☐ Dr. Aiman El-Maleh	☐ Dr. Muhamed Mudawar
☐ Dr. Ali Al-Suwaiyan	☐ Dr. AbdulAziz Tabakh

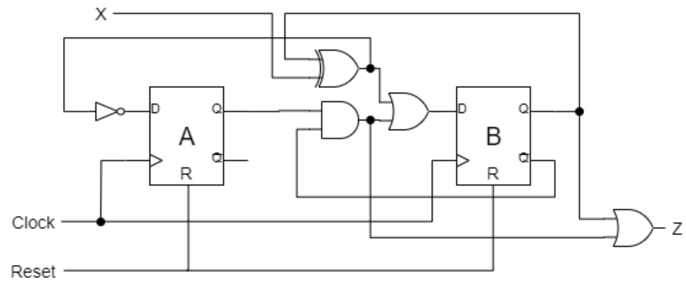
Notes:

- Do not open the exam book until instructed
- **No Calculators are allowed** (*basic, advanced, cell phones, etc.*)
- Answer all questions
- All steps must be shown
- Any assumption made must be clearly stated

Question	Maximum Points	Your Points
Q1	9	
Q2	15	
Q3	13	
Q4	7	
Q5	16	
Total	60	

Question 1: **(9 points)**

The sequential circuit shown below has a single input X and a single output Z. Answer the following questions regarding this circuit.



(i) (1 point) The circuit is __________ (Mealy/Moore) machine.

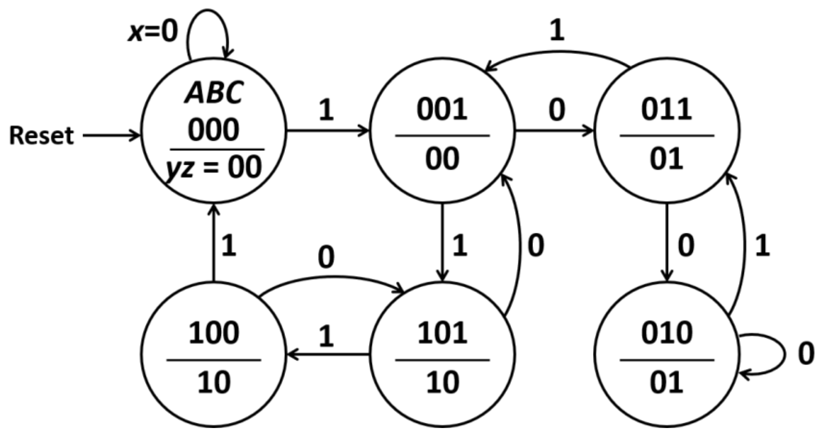
(ii) (3 points) Derive the Boolean equations for the inputs of the flip-flops (DA, DB) and the output (Z).

(iii) (4 points) Show the state table of this circuit and draw the corresponding state diagram.

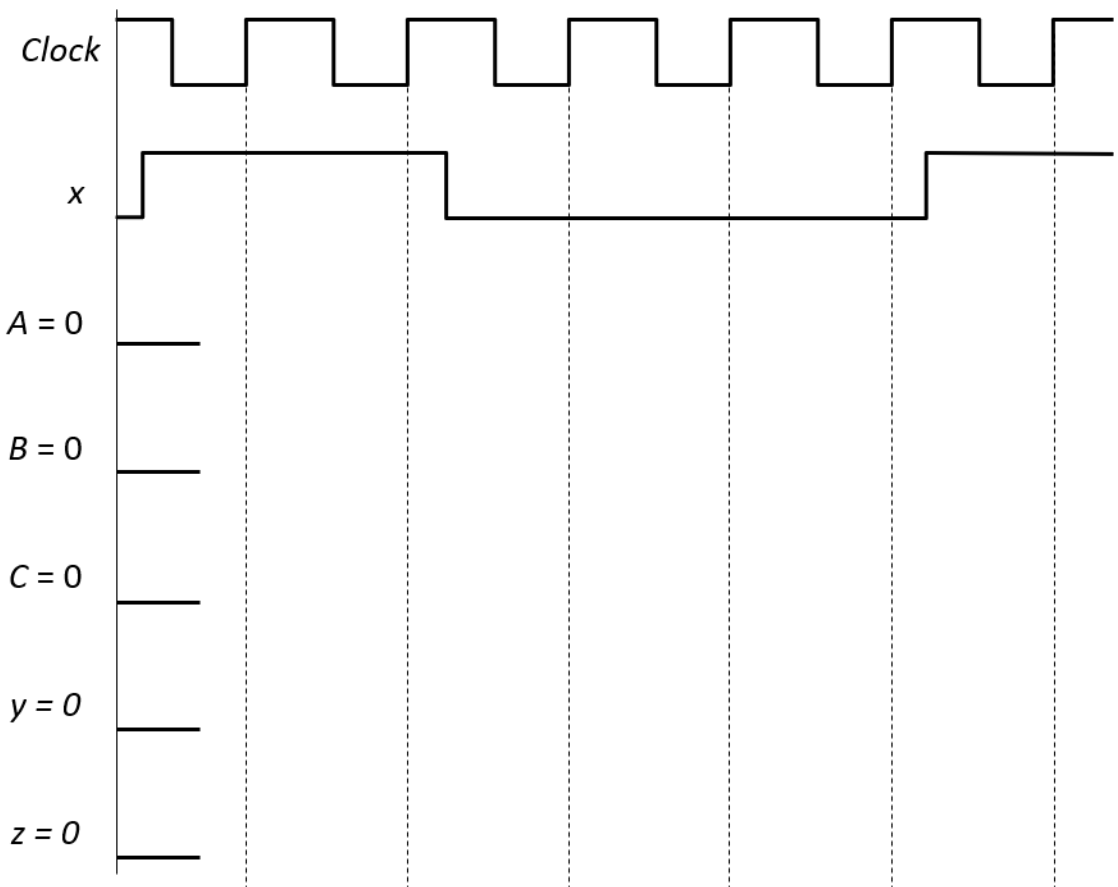
(iv)(1 point) Identify all unused states, if any, and explain your answer.

Question 2. **(15 points)**

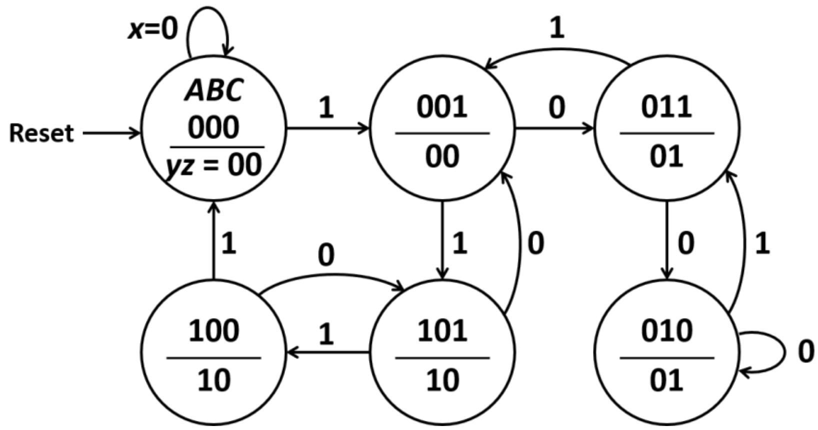
The following state diagram is for a sequential circuit with input *x*, two outputs *y* and *z*, three state variables *A*, *B*, *C*, **positive** edge-triggered D flip-flops, and **synchronous** reset.



(i) (**5 points**) Complete the timing diagram for the above state diagram, showing the waveforms of the state variables *A*, *B*, *C*, and the outputs *y* and *z*. Initially, the flip-flops are reset to 0.



(ii) (**6 points**) Write a behavioral description of the state diagram in Verilog. The next-state logic should be described directly from the state diagram. For the unused states that do not appear in the state diagram, assume that they always make a transition to the start state, regardless of the input *x*. The output of all unused states should be *yz* = 11.



(iii) (**4 points**) Using K-maps, obtain the minimized next-state equation at the input of flip flop *C* and for the output *y*. For the unused states, assume that they always make a transition to the start state, regardless of the input *x*. The output of all unused states should be *yz* = 11.

Question 3. **(13 points)**

(i) **(6 points)** It is required to design a sequential circuit that receives a serial input X and produces two serial outputs Y1 and Y0. The circuit is to be designed to detect the number of 1's in only one of the following input sequences 10, 110, 1110. The input/output relation is shown in the table given below:

Input Sequence	# of 1's (Y1Y0)
10	01
110	10
1110	11
11110 or any other sequence 111..110	00

Draw the state diagram for this circuit with minimum number of states assuming **Mealy** model. Assume that the circuit has a *Reset* input when asserted resets the machine to the reset state. The following is an example of an input/output stream, starting at the initial state:

Clock Cycle	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Input X	1	0	0	1	1	1	0	1	1	0	1	1	1	1	1	0
Output Y1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
Output Y0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0

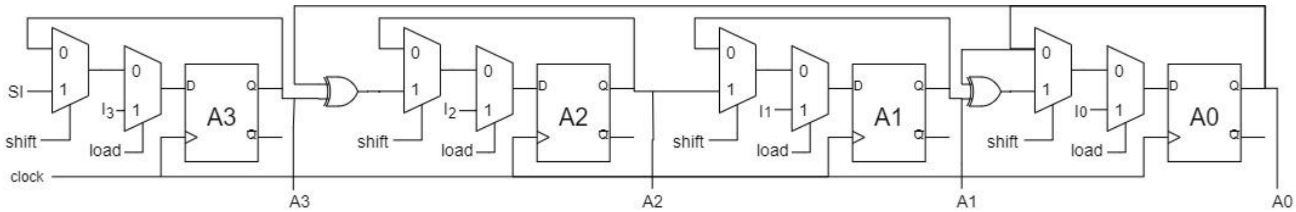
(ii) (7 points) It is required to design a sequential circuit that receives a serial input X and produces a serial output Y. The circuit is to be designed to set Y to 1 when it detects one of the sequences 101 or 010, with sequence overlapping. Draw the state diagram for this circuit with minimum number of states assuming **Moore** model. Assume that the circuit has a *Reset* input when asserted resets the machine to the reset state.

The following is an example of an input/output stream, starting at the initial state:

Clock Cycle	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Input X	0	1	0	1	1	1	0	1	1	0	0	1	0	0	1	1
Output Y	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0

Question 4: **(7 points)**

(i) **(3 points)** For the circuit shown below, assume the initial state (A3, A2, A1, A0) = 0000.



Fill in the entries of the table given below for the given sequence of inputs. Indicate the flip-flops values following each clock pulse. Assume that the indicated values for input signals are valid and stable before the edge of the clock. Also note that the flip-flops values, A3-A0, shown at a given row for Clock#i indicate the values of flip-flops after pulsing Clock i.

Clock pulse #	Load	Shift	I3 – I0	SI	A3-A0
1	1	1	0110	1	0110
2	0	1	1100	1	1011
3	0	1	1010	1	1000
4	0	0	0011	1	
5	0	1	0000	0	
6	0	1	1111	1	

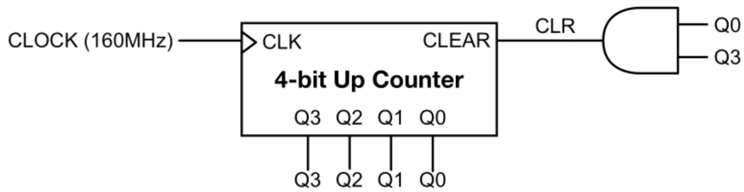
(i) **(4 points)** Consider a 3-bit synchronous shift register that takes a 2-bit function selector, S1 and S0, with the following functionality

S1S0	Operation
00	No change (A2<=A2, A1<=A1, A0<=A0)
01	Shift left (A2<=A1, A1<=A0, A0<=SI)
10	Shift right (A2<=SI, A1<=A2, A0<=A1)
11	Rotate right (A2<=A0, A1<=A2, A0<=A1)

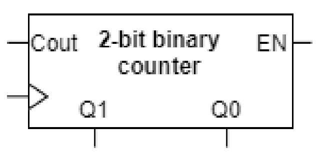
Show an implementation of this shift register using only positive-edge triggered D-FFs and muxes. Clearly label all your components.

Question 5. **(16 points)**

(i) (**2 points**) Given the following 4-bit synchronous up-counter with synchronous clear signal, determine the frequencies of the two signals Q_0 and CLR. Assume that the counter is initialized to 0. The frequency of the input clock is 160 MHz.



(ii) (**2 points**) Assume that you have the following 2-bit synchronous counter with count enable (EN) and carry out (Cout) signal. In this counter, the carry out signal is defined according to this equation Cout $= Q_0 \cdot Q_1 \cdot$ EN. The following symbol shows the input and output ports of this counter:



Show how to implement a 6-bit synchronous counter using the minimum number of this 2-bit counter components and any needed additional gates. Clearly label all input and output signals.

(iii) (**12 points**) Consider a 3-bit synchronous counter that has an asynchronous reset and 2-bit function selector, F1 and F0, with the following functionality:

F1	F0	Function
0	0	No change
0	1	Counts up by 1
1	0	Counts down by 1
1	1	Counts up by 2

(a) (**7 points**) Show an implementation of this counter using only positive-edge triggered D-FFs, muxes, and logic gates. Do not use adders. Clearly label all your components.

(b) (**5 points**) Show a behavioral Verilog module that models this counter.