University of Windsor

School of Computer Science
Faculty of Science

COMP-2650: Computer Architecture I: Digital Design
Fall 2020

Assignment#	Date	Title	Due Date	Grade Release Date
Lec 10	Nov 30- Dec 02, 2020	**W12: Sequential Logic**	Dec. 02, 2020 + Extension Wednesday Midnight AoE	Dec. 23, 2020

The lecture (weekly) assignments are to practice on topics covered in the lectures and improve the student's critical thinking and problem-solving skills in ad hoc topics closely related but not covered in the lectures. Lecture assignments also help students with research skills, including accessing, retrieving, and evaluating information (information literacy).

Deadline: According to the school's policy, no assignment can be due on the last week of class or after. Also, we cannot remove the assignments for the last two weeks since it changes the course outline for the marking schema, which is not possible at this time of the semester. This conflict of policies happens because we extend the labs' deadlines for two weeks to reduce the workload. So, here is the solution:

- The official due date is what is mentioned above: Dec. 02, 2020, Wednesday Midnight AoE.
- There is an extension of 2 weeks to the official deadline for all students. Therefore, we accept submission till Dec. 16, 2020, Wednesday Midnight AoE.
- The difficulty of this week's assignment is in easy level (selecting only 1 question) that students can do the assignments within the official due date.

Lecture Assignments Deliverables

You should answer one of the below questions based on your preference using an editor like MS Word, Notepad, and the likes or pen in papers. You have to write and scan the papers clearly and merge them into a single file in the latter case. In the end, you have to submit all your answers in one single pdf file `COMP2650_Lec10_{UWinID}.pdf` containing the following items:

1. Your name, UWinID, and student number
2. The question Id for each answer. Preferably, the questions should be answered in order of increasing Ids. *Please note that if your answers cannot be read, you will lose marks.*
3. Including the questions in your submission pdf file is optional.

Please follow the naming convention as you lose marks otherwise. Instead of {UWinID}, use your own UWindsor account name, e.g., mine is hfani@uwindsor.ca, so, my submission would be: `COMP2650_Lec10_hfani.pdf`

Lecture Assignments
(select only 1 question based on your preference)

A **state equation** (also called a transition equation) specifies the next state as a function of the present state and inputs (if any). For instance, given the following state table,

Input	Q(T)		Q(T+1)	
X	B	A	B	A
0	0	0	1	1

0	0	1	1	1
0	1	0	X	0
0	1	1	0	0
1	0	0	0	1
1	0	1	0	1
1	1	0	X	0
1	1	1	1	0

$Q(T+1)_A$ or simply A(T+1) is equal to sum of those minterms that makes the next state 1 based on the current states of all other memory units and the inputs:

$$Q(T+1)_A = A(T+1) = F(X, B, A) = \sum(0,1,4,5)$$

Similarly, for B,

$$Q(T+1)_B = B(T+1) = F(X, B, A) = \sum(0,1,7) + d(2,6)$$

So, we have:
(a) Output equation: the sum of minterms that make the *output* 1 (or product of MAXTERMs that make the output 0)
(b) Input (excitation) equation: the sum of minterms that make *the input to the flip-flop* 1 (or product of MAXTERMs that make the input to the flip-flop 0) based on an action required for the flip-flop
(c) State equation: the sum of minterms that make the *next state* 1 (or product of MAXTERMs that make the next state 0)

They all follow the same principle: writing the equation (Boolean function) based on the current state and the input variables. In the following questions, you might be given or asked to write the state equations for each flip-flop.

1. A PN flip-flop has four operations: clear to 0, no change, complement, and set to 1, when inputs P and N are 00, 01, 10, and 11, respectively. Design PN flip-flop using:

   (a) JK-FF
   (b) RS-FF
   (c) D-FF
   (d) T-FF

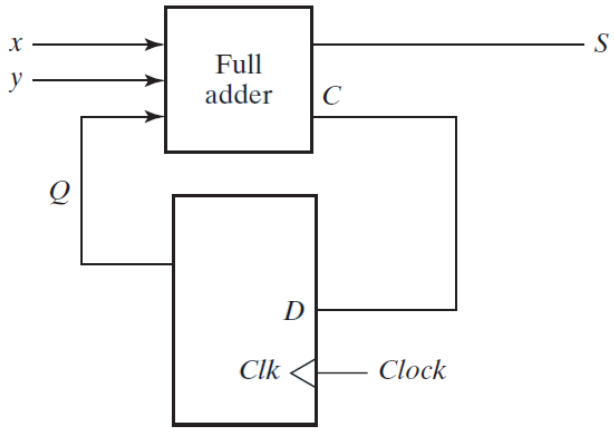
   (*Hint: this design problem can be seen as the design for a sequential circuit with 1 memory unit, 2 inputs (X=P, Y=N) and two outputs (F_1=Q and F_2=F_1'=Q'.*)

2. A sequential circuit with two D flip-flops A and B, two inputs, x and y; and one output z is specified by the following next-state equations (look at the beginning of the assignments for what the state equation is) and output equations:

$$A(t+1) = xy' + xB$$
$$B(t+1) = xA + xB'$$
$$z = A$$

   (a) List the state table for the sequential circuit.
   (b) Draw the corresponding state diagram.
   (c) Draw the logic diagram of the circuit.

3. A sequential circuit has two inputs x and y, and one output S. It consists of a full-adder circuit connected to a D flip-flop, as shown below. Derive the state table and state diagram of the sequential circuit.



4. A sequential circuit has two JK-FFs A and B and one input x. The circuit is described by the following input (excitation) equations:

$$J_A = x;\ K_A = B$$
$$J_B = x;\ K_B = A'$$

(a) Draw the state table of the circuit.
(b) Draw the state diagram of the circuit.

5. A sequential circuit has two JK flip-flops A and B, two inputs x and y, and one output z. The flip-flop input equations and circuit output equation are:

$$J_A = Bx + B'y';\ K_A = B'xy'$$
$$J_B = A'x;\ K_B = A + xy'$$
$$z = Ax'y' + Bx'y'$$

(a) Is this circuit a Moore model or Mealy model? Explain your answer.
(b) Draw the logic diagram of the circuit.
(c) Tabulate the state table.
(c) Derive the state equations for A and B (look at the beginning of the assignments for the state equation).

6. For the following state table

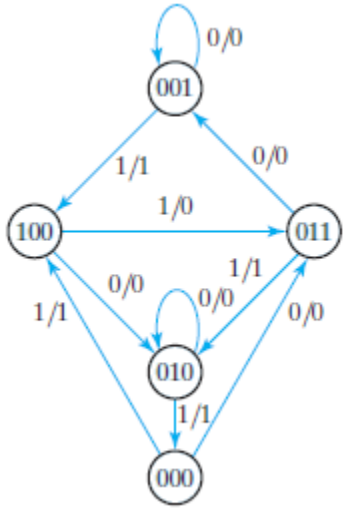
Present State	Next State		Output	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1

h	g	a	1	0

Draw the corresponding state diagram.

7. A sequential circuit has three flip-flops A, B, C; one input x_{in} and one output y_{out}. The state diagram is shown below. The circuit is to be designed by treating the unused states as don't-care conditions. Analyze the circuit obtained from the design to determine the effect of the unused states.

   (a) Use D-FF in the design.
   (b) Use JK-FF in the design.



(*Hint: as seen, some states (nodes) are missing, such as 101, 110 and 111. If we reach at these values in the memory units at some point, the next state is not determined, that is, don't cater conditions. When writing the input equations and Boolean function for the output, you can assume any binary value inside the memory unit. It helps with more simplification.*)

8. Design the sequential circuit specified by the state diagram below, using T-FF.