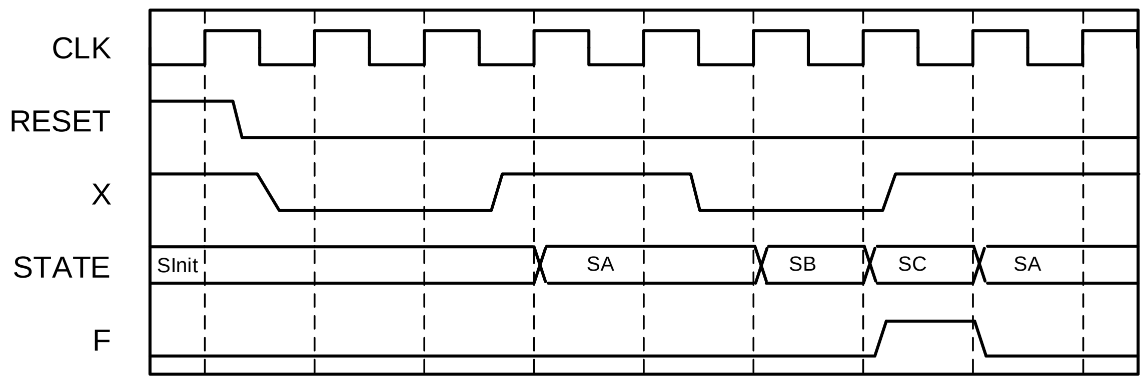
EE 209 Homework 3

Name: __Solutions_______________________________
Due: **Score: ________**

1.

The waveform is below:



2.

(a) There is only one D Flip flop, and Q_1 can be 0 or 1. Thus, there are 2 states for this circuit.

(b) This is a Mealy Machine. Because the output F is the function of the current state and the external input A.

(c) The maximum number of transition arrows originating from one state is 8. Because we have A, B and C three external inputs, and there are 8 possible combinations of one 3-bits binary number.

(d) No. Because when A is equal to 1, the input of the D Flip flop is definitely 0. Thus, when the next clock tick comes, Q_1 is equal to 0 and F is 0 too.

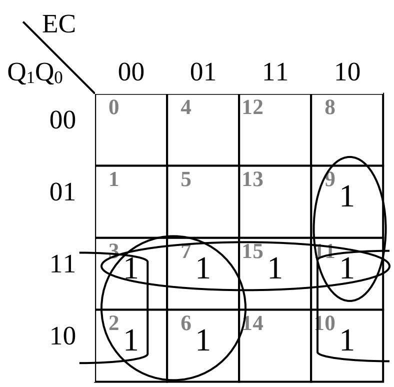
3.

Because we have 4 states in the state diagram, we use two D flip-flops to implement state memory. We design the combinational logic for the next-state logic and the output function logic by building up the state/transition table (with E=ENTER, C=CORRECT).

Current State		Next State								Output
		E C = 0 0		E C = 0 1		E C = 1 0		E C = 1 1		
Sym.	Q1Q0	Sym.	Q1*Q0*	Sym.	Q1*Q0*	Sym.	Q1*Q0*	Sym.	Q1*Q0*	Alarm
OFF	00	OFF	00	OFF	00	OFF	00	MON	01	0
MON	01	MON	01	MON	01	1WR	10	OFF	00	0
AL	11	AL	11	AL	11	AL	11	AL	11	1
1WR	10	1WR	10	1WR	10	AL	11	OFF	00	0
			D1 D0		D1 D0		D1 D0		D1 D0	

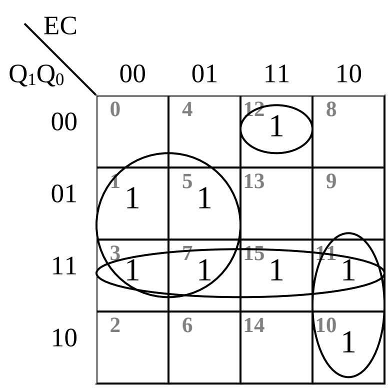
After we get the state/transition table, we build the K-Maps to simplify expressions for D1, D0, and Alarm.

- D_1 K-Map:



$D_1 = Q_1Q_0 + ENTER'Q_1 + CORRECT'Q_1 + CORRECT'\cdot ENTER\cdot Q_0$

- D_0 K-Map:



$D_0 = Q_1Q_0 + ENTER'\cdot Q_0 + ENTER\cdot CORRECT'\cdot Q_1 + E\ C\ Q_1'\ Q_0'$

- Alarm K-Map:

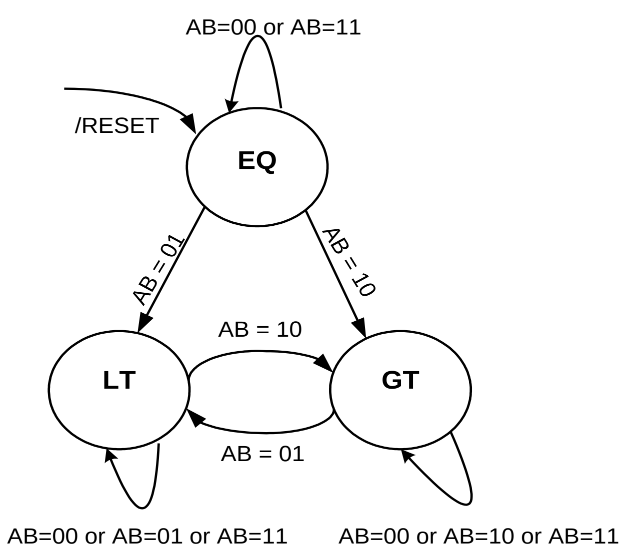
Q_1Q_0 \ EC	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3: 1	7: 1	15: 1	11: 1
10	2	6	14	10

Alarm = Q_1Q_0

- The initial state is OFF which has the code Q1Q0=00. Thus we want both flip-flops to initialize to 0. To do this, connect RESET to the CLR inputs and tie the SET/PRE inputs to GND (never want to initialize to 1).

4. **(a.)**

(1) First, we build the state diagram.

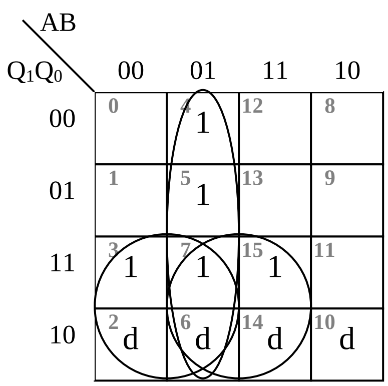


(2) Because we have 3 states in the state diagram, we use two D flip-flops to implement state memory. We design the combinational logic for the next-state logic and the output function logic by building up the state/transition table.

Curr. State		Next State								Output		
		A B = 00		A B = 01		AB = 11		AB = 10				
Sym.	Q_1Q_0	Sym.	$Q_1^*Q_0^*$	Sym.	$Q_1^*Q_0^*$	Sym.	$Q_1^*Q_0^*$	Sym.	$Q_1^*Q_0^*$	EQ	LT	GT
GT	0 0	GT	0 0	LT	1 1	GT	0 0	GT	0 0	0	0	1
EQ	0 1	EQ	0 1	LT	1 1	EQ	0 1	GT	0 0	1	0	0
LT	1 1	LT	1 1	LT	1 1	LT	1 1	GT	0 0	0	1	0
X	1 0	X	d d	X	d d	X	d d	X	d d	X	X	X
			D_1 D_0		D_1 D_0		D_1 D_0		D_1 D_0			

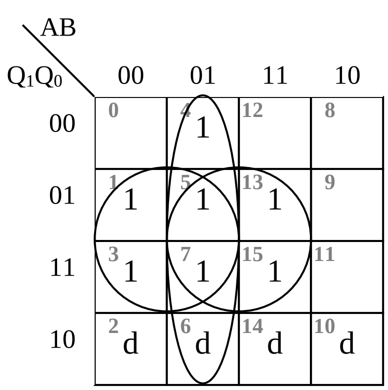
(3) Build the K-Maps to simplify expressions for D1, D0, EQ, LT and GT.

- D_1 K-Map:



$D_1 = A'B + A'Q_1 + BQ_1$

- D_0 K-Map:



$D_0 = A'B + A'Q_0 + BQ_0$

- EQ K-Map:

  $EQ = Q_1'Q_0$ or   EQ = Q1 xor Q0

- LT K-Map:

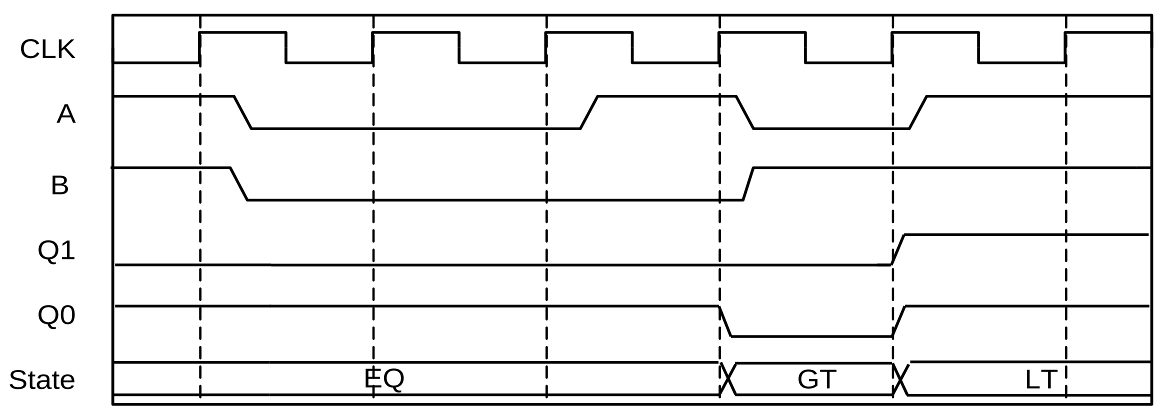
  $LT = Q_1$

- GT K-Map:

  $GT = Q_0'$

- The initial state is EQ which has the code Q1Q0=01. Thus we want Q1 to initialize to 0. To do this, connect RESET to the CLR input and tie the SET/PRE input to GND (never want to initialize to 1). However we want Q0 to initialize to 1. To do this, connect RESET to the PRE/SET input and tie the CLR input to GND (never want to initialize to 0).

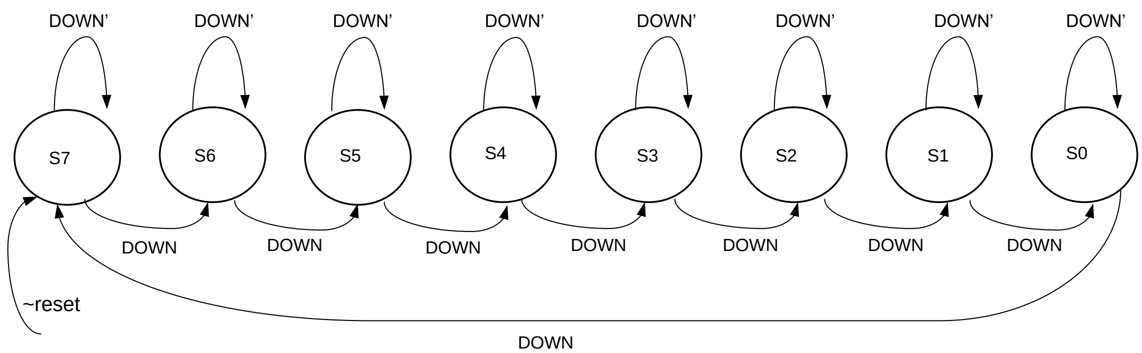
The waveform is below:



5. **(a.) We need 8 states and each of them stores one 3-bit binary number ranging from 000 to 111.**

State	S0	S1	S2	S3	S4	S5	S6	S7
Assignment (Q_2 Q_1 Q_0)	000	001	010	011	100	101	110	111

The state diagram is below:



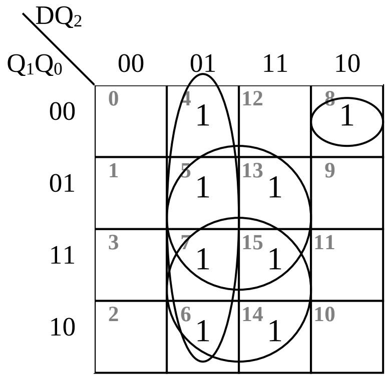
(b) Because we have 8 states in the state diagram, we use three D flip-flops to implement state memory.

We design the combinational logic for the next-state logic and the output function logic by building up the state/transition table.

Current State		Next State / Output (Q_i is the output)			
		DOWN = 0		DOWN = 1	
Symbol	Q2 Q1 Q0	Symbol	Q2* Q1* Q0*	Symbol	Q2* Q1* Q0*
S0	0 0 0	S0	0 0 0	S7	1 1 1
S1	0 0 1	S1	0 0 1	S0	0 0 0
S2	0 1 0	S2	0 1 0	S1	0 0 1
S3	0 1 1	S3	0 1 1	S2	0 1 0
S4	1 0 0	S4	1 0 0	S3	0 1 1
S5	1 0 1	S5	1 0 1	S4	1 0 0
S6	1 1 0	S6	1 1 0	S5	1 0 1
S7	1 1 1	S7	1 1 1	S6	1 1 0
			D2 D1 D0		D2 D1 D0

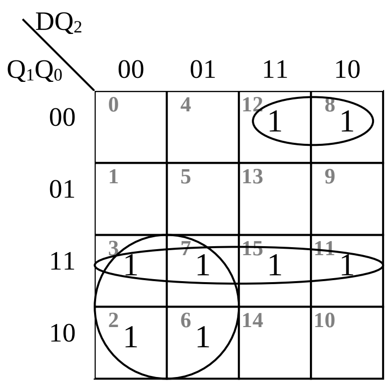
After we get the state/transition table, we build the K-Maps to simplify expressions for D2, D1, D0.

- D_2 K-Map:



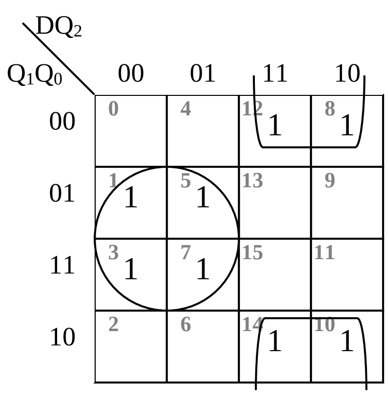
$D_2 = D'Q_2 + Q_2Q_0 + Q_2Q_1 + DQ_2'Q_1'Q_0'$

- D_1 K-Map:



$D_1 = Q_1Q_0 + D'Q_1 + DQ_1'Q_0'$

- D_0 K-Map:



$D_0 = D'Q_0 + DQ_0'$

- The initial state is 111 which has the code Q2Q1Q0=111. Thus we want all flip-flops to initialize to 1. To do this, connect RESET to the PRE/SET inputs and tie the CLR inputs to GND (never want to initialize to 0).

(c) The waveform is below: