1. Design a modulo-6 counter, which counts 0,1,2,3,4,5,0,1,..... The counter counts the clock pulses if its enable input,w, is equal to 1. Use D flip flops in your circuit. If the circuit ever finds itself in an unused state (6 or 7), it should transition to state 0 with the next clock trigger to avoid being stuck in an unused state. Use a formal design procedure.

Drawing the state table for the given modulo 6 counter.

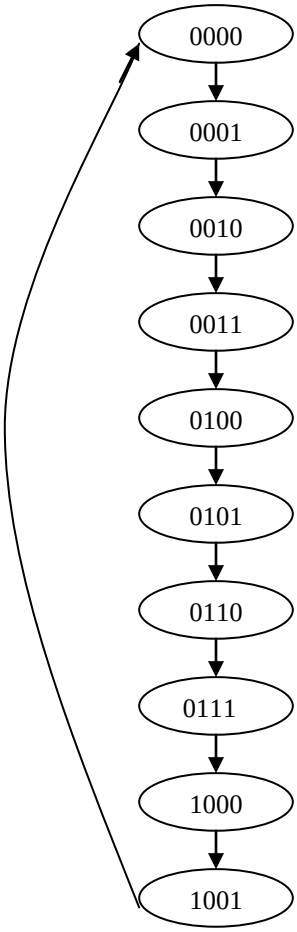
Present state			**Input**	**Next state**		
Q2	**Q1**	**Q0**	**W**	**Q2+**	**Q1+**	**Q0+**
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	0
0	1	0	1	0	1	1
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	1	0	0
1	0	0	1	1	0	1
1	0	1	0	1	0	1
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0

For D flip-flops, the input into the flip-flops is the same as the next state that are shown in the table. (D2=Q2+, D1 = Q1+, D0 = Q0+).

Now drawing 4-variable K maps for D flip flop inputs D2, D1 and D0 we get the following result (K-maps not shown):
D2 = Q2Q1’Q0’ + Q2Q1’W’ + Q2’Q1Q0W
D1 = Q2’Q1Q0’ + Q2’Q1W’ + Q2’Q1’Q0W
D0 = Q2’Q0’W + Q1’Q0’W + Q2’Q0W’ + Q1’Q0W’.
The above equations completely define the required circuit.

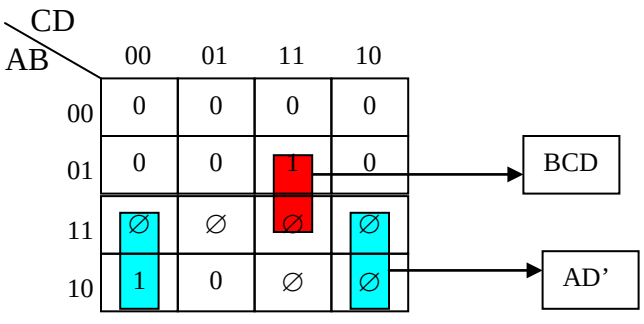
2. Consider the design of a 4-bit BCD counter that counts in the following way: 0000, 0001, 0010, 0011,…., 1001 and back to 0000.

A) Draw the state diagram and next state table.

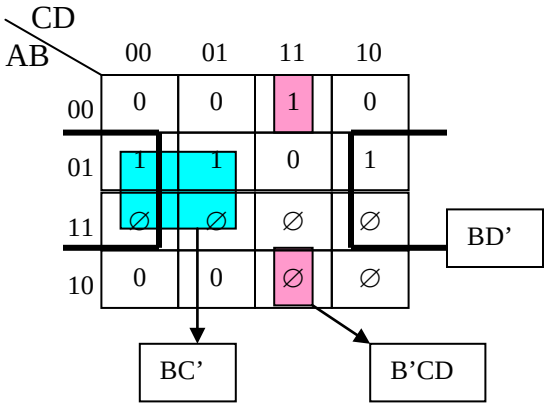


Present Stage				Next Stage				D Flip Flop Input			
A	**B**	**C**	**D**	**A+**	**B+**	**C+**	**D+**	**Da**	**Db**	**Dc**	**Dd**
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0

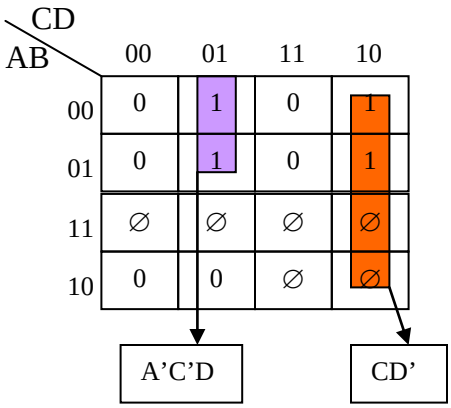
B) Implement the counter using D flip-flops.



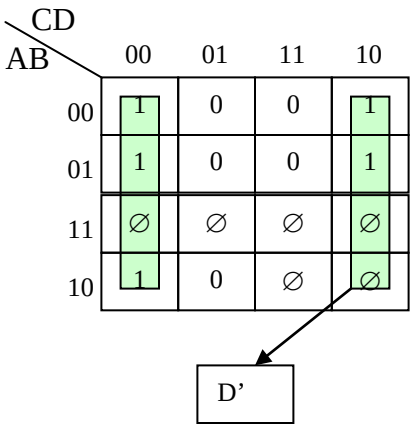
Da = AD' + BCD



Db = BC' + B'CD + BD'



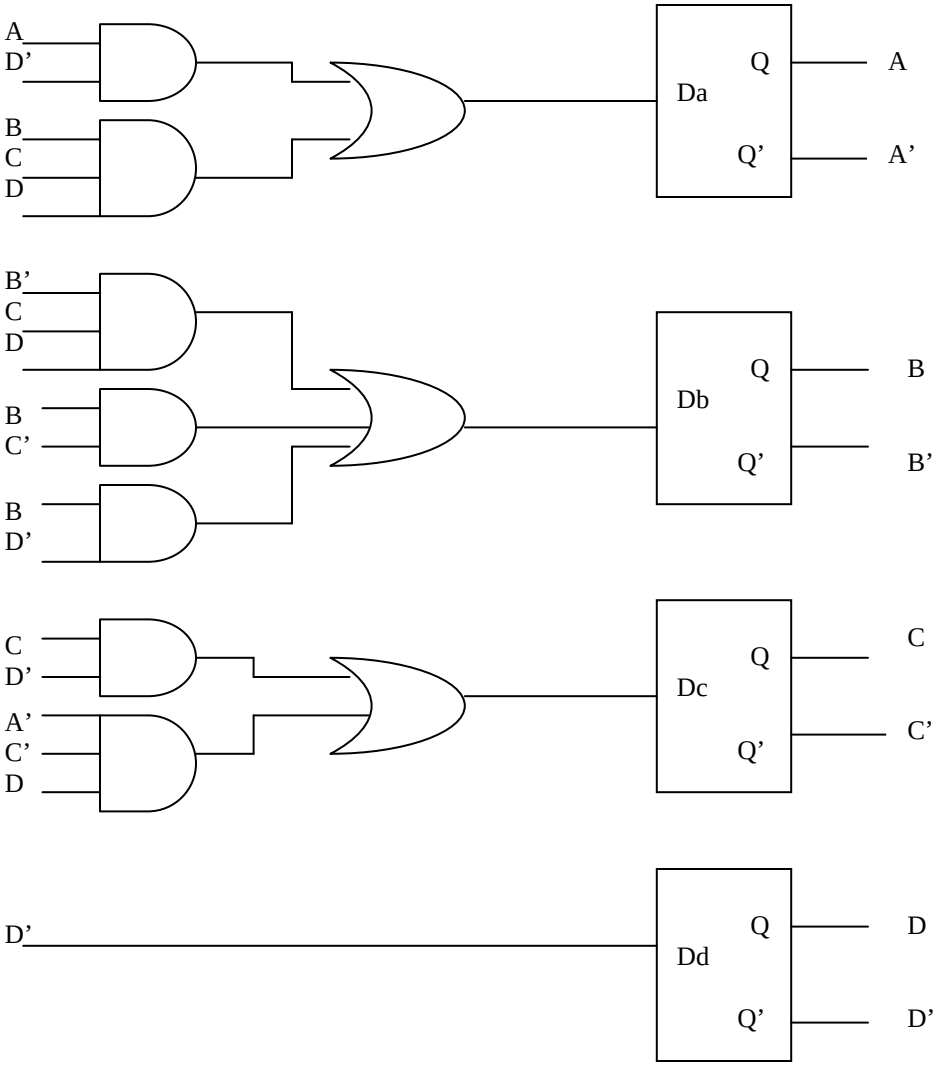
Dc = CD' + A'C'D



Dd = D'

Note that ∅S represent don't cares.

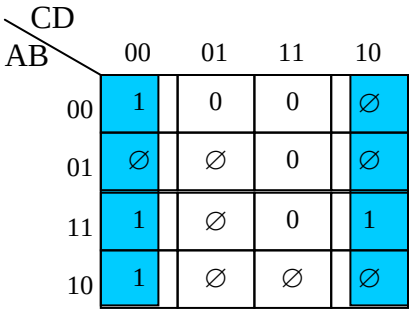
Logic circuit



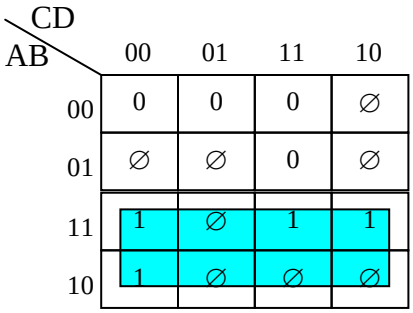
3. The 4-bit Johnson counter advances thru the sequence: 0000, 1000, 1100, 1110,1111, 0111, 0011, 0001 and repeat.

   A) Implement this counter using D flip-flops.

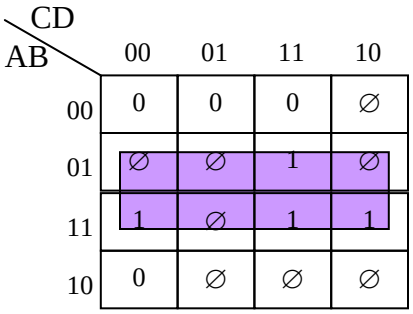
Present Stage				**Next Stage**				**D Flip Flop Input**			
A	**B**	**C**	**D**	**A+**	**B+**	**C+**	**D+**	**Da**	**Db**	**Dc**	**Dd**
0	0	0	0	1	0	0	0	1	0	0	0
1	0	0	0	1	1	0	0	1	1	0	0
1	1	0	0	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	1	0	0	1	1
0	0	1	1	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0



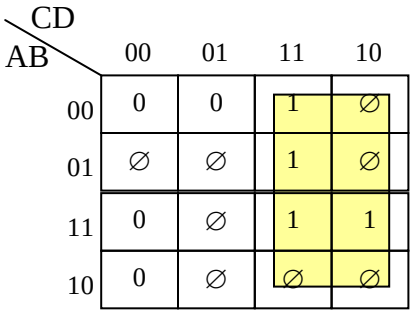
Da = D’



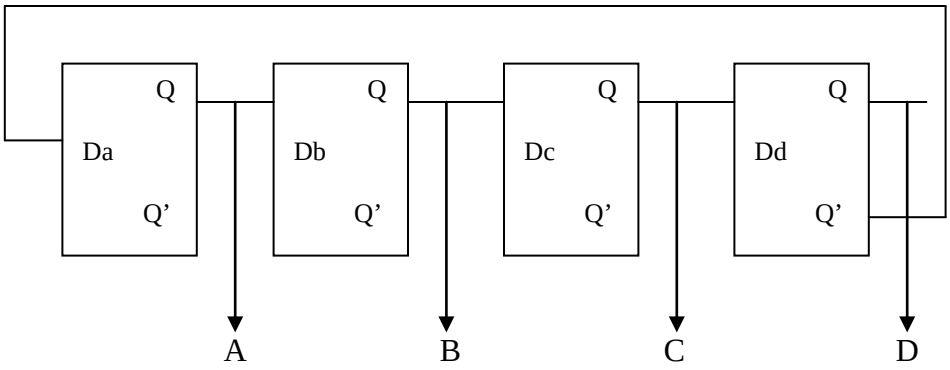
Db = A



Dc = B

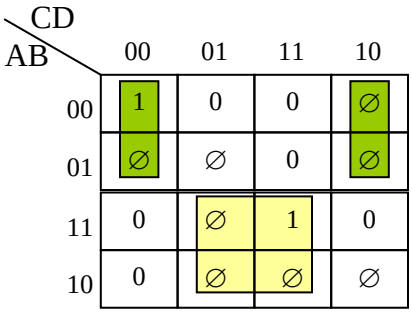


Dd = C

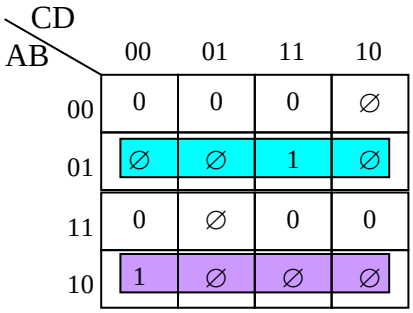


B) Implement this counter using T flip-flops.

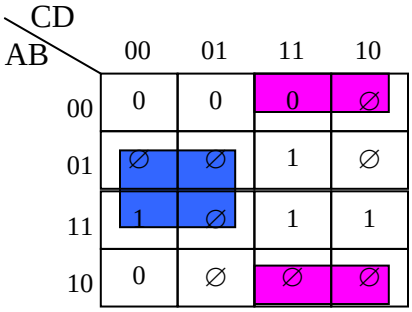
Present Stage				Next Stage				T Flip Flop Input			
A	**B**	**C**	**D**	**A+**	**B+**	**C+**	**D+**	**Ta**	**Tb**	**Tc**	**Td**
0	0	0	0	1	0	0	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1	0	0
1	1	0	0	1	1	1	0	0	0	1	0
1	1	1	0	1	1	1	1	0	0	0	1
1	1	1	1	0	1	1	1	1	0	0	0
0	1	1	1	0	0	1	1	0	1	0	0
0	0	1	1	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	1



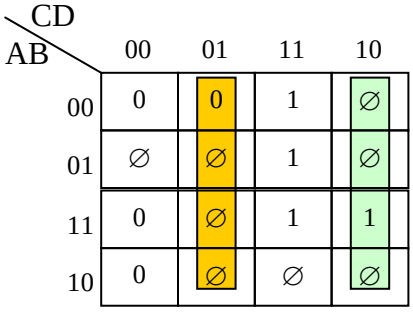
Ta = A'D' + AD



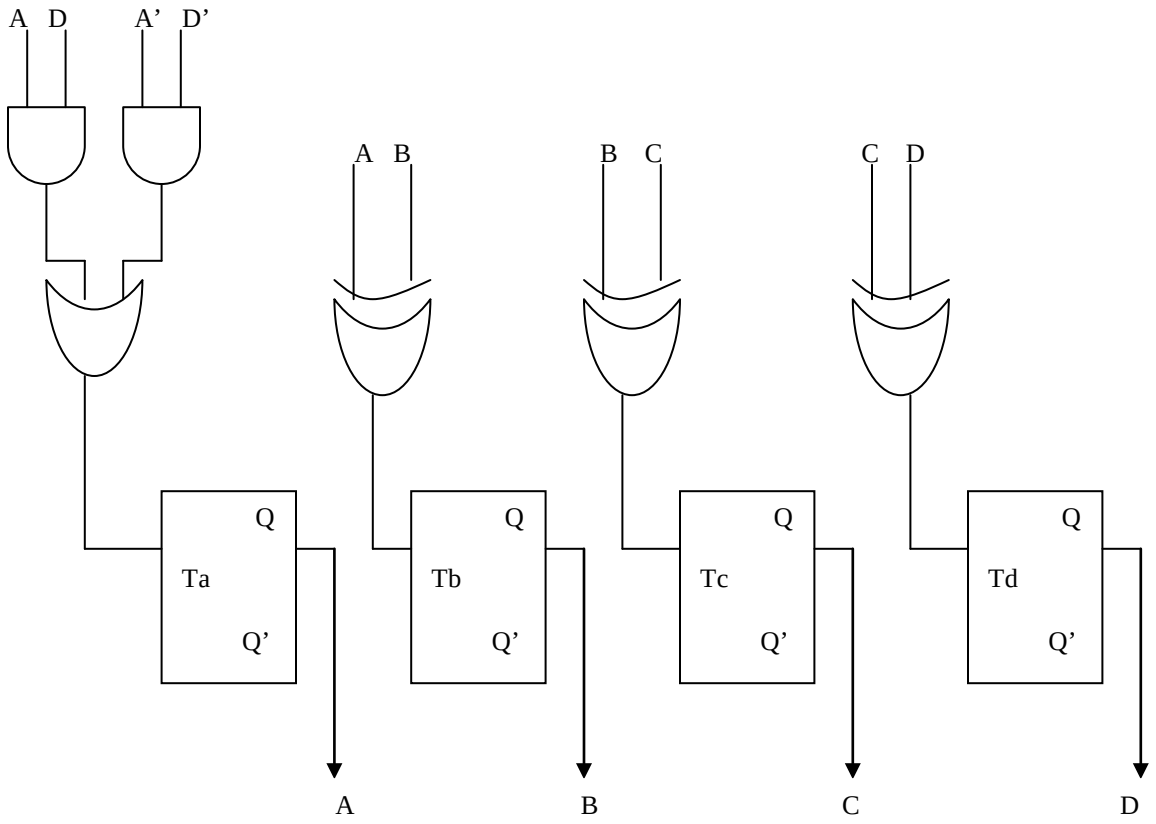
Tb = A'B + AB'



Tc = BC' + B'C



Td = C'D + CD'

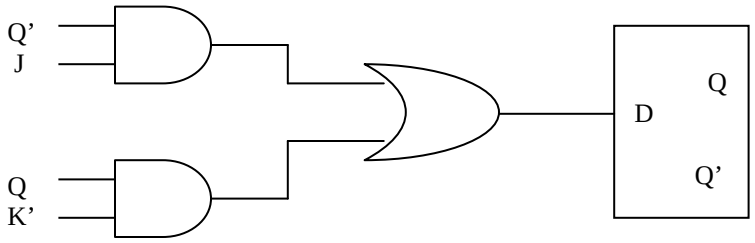

A D
A' D'
A B
B C
C D
Q
Ta
Q'
Q
Tb
Q'
Q
Tc
Q'
Q
Td
Q'
A
B
C
D

4. Show how to implement a JK flip-flop with a D flip-flop.

J	K	Q	Q+	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Q \ JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

D = Q'J + QK'

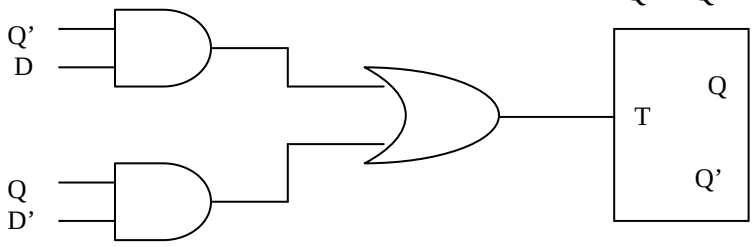


5. Show how to implement a D flip-flop with a T flip-flop.

D	Q	Q+	T
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Q \ D	0	1
0	0	1
1	1	0

T = DQ' + QD'



6. Show how to implement a JK flip-flop with a T flip-flop.

J	K	Q	Q+	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Q \ JK	00	01	11	10
0	0	0	1	1
1	0	1	1	0

T = Q'J + QK