

Date:

21-04-2020

## CS226 - Switching Theory Lab [10]

Name: M. Maheeth Reddy

Topic: Sequential Design

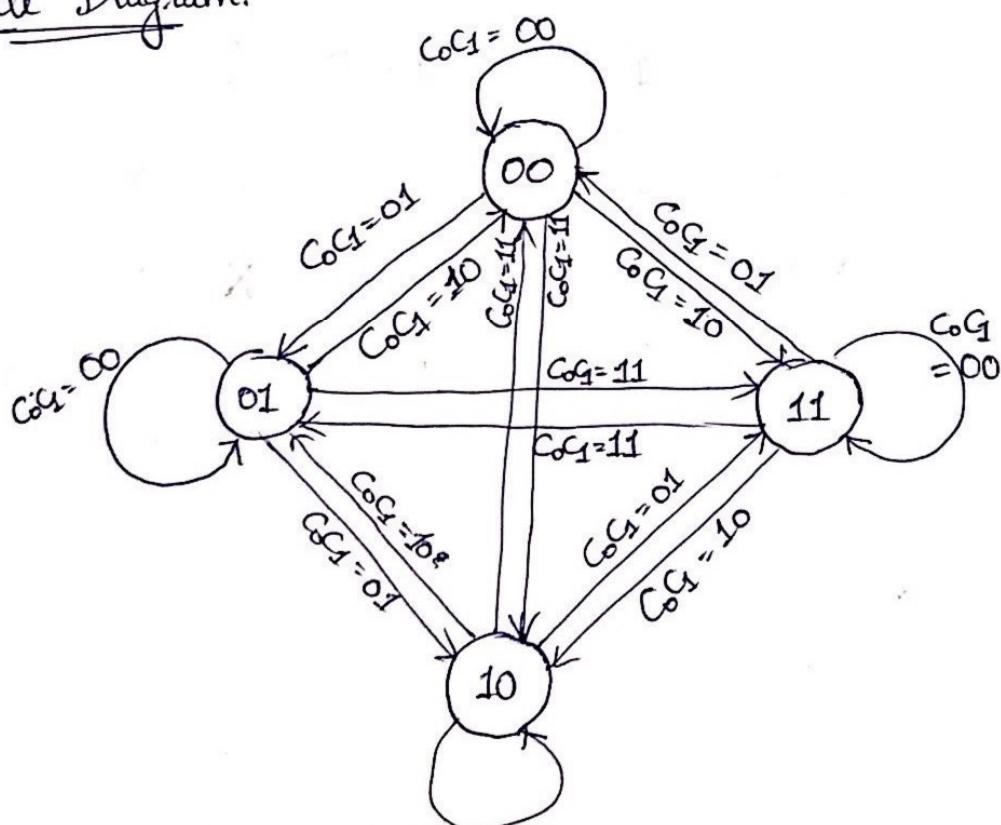
Roll No.: 1801CS31

Ans 1

State Table of given circuit:

$C_0$	$G_1$	State
0	0	Stop Counting
0	1	Count up by one
1	0	Count down by one
1	1	Count by two

State Diagram:



I) Implementation using SR FlipFlop:

Circuit Excitation Table:

$C_0$	$C_1$	$Q1_{in}$	$Q0_{in}$	$Q1_{out}$	$Q0_{out}$	$S1$	$R1$	$S0$	$R0$
0	0	0	0	0	0	0	X	0	X
0	0	0	1	0	1	0	X	X	0
0	0	1	0	1	0	X	0	0	X
0	0	1	1	1	1	X	0	X	0
0	1	0	0	0	1	0	X	1	0
0	1	0	1	1	0	1	0	0	1
0	1	1	0	1	1	X	0	1	0
0	1	1	1	0	0	0	1	0	1
1	0	0	0	1	1	1	0	1	0
1	0	0	1	0	0	0	X	0	1
1	0	1	0	0	1	0	1	1	0
1	0	1	1	1	0	X	0	0	1
1	1	0	0	1	0	1	0	0	X
1	1	0	1	1	1	1	0	X	0
1	1	1	0	0	0	0	1	0	X
1	1	1	1	0	1	0	1	X	0

K-Maps for S1, R1, S0, R0

		Q1 Q0	00	01	11	10
		C0 C1	00	X	X	
		00	00	1 1		
		01				
		11	X		X	
		10		1 1		

$$R_0 = \overrightarrow{Q_{0n}} [C_0 \oplus C_1]$$

		Q1 Q0	00	01	11	10
		C0 C1	00	X	X	
		00	00	1	1	
		01				
		11		X	X	
		10	D		1	

$$S_0 = \overrightarrow{Q_{0n}} [C_0 \oplus C_1]$$

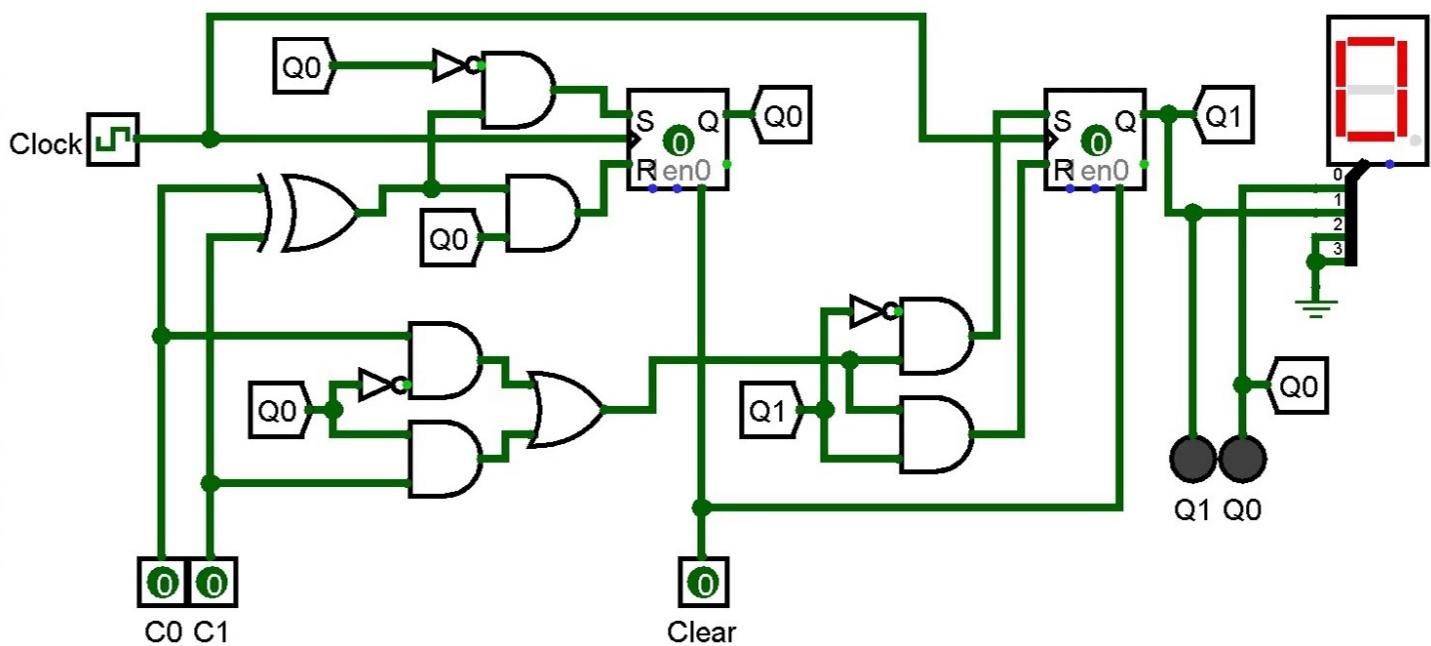
		Q1 Q0	00	01	11	10
		C0 C1	00	X	X	
		00	00	1		
		01	X			
		11		1	1	
		10	X		1	

$$R_1 = \overrightarrow{Q_{1n}} (C_1 Q_{0n} + C_0 \overline{Q_{0n}})$$

		Q1 Q0	00	01	11	10
		C0 C1	00	X	X	
		00	00	1		
		01		1		
		11	1	1		
		10	1		X	

$$S_1 = \overrightarrow{Q_{1n}} (C_1 Q_{0n} + C_0 \overline{Q_{0n}})$$

Logic Diagram:

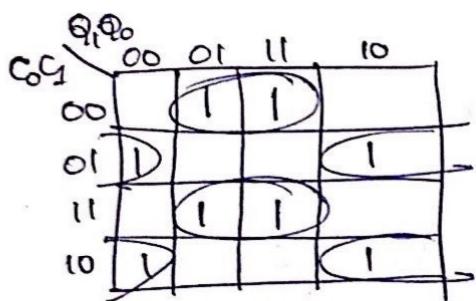


## II) Implementation using D, flipflop:

Circuit Excitation Table

$C_0$	$C_1$	$Q_{1n}$	$Q_{0n}$	$Q_{1n+1}$	$Q_{0n+1}$	$D_1$	$D_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1
0	0	1	0	1	0	1	0
0	0	1	1	1	1	1	1
0	1	0	0	0	1	0	1
0	1	0	1	1	0	1	0
0	1	1	0	1	1	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	1	0	1
1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	0
1	1	0	1	0	1	1	1
1	1	1	0	0	0	0	0
1	1	1	1	0	1	0	1

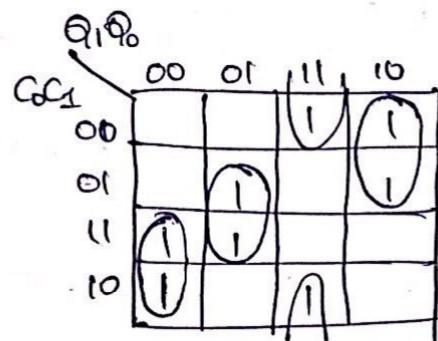
## K-Maps for D1, D0 :



$$D_0 = \left\{ \begin{array}{l} \bar{C}_0 \bar{C}_1 Q_{0n} + \bar{C}_0 C_1 \bar{Q}_{0n} \\ + C_0 \bar{C}_1 \bar{Q}_{0n} + C_0 C_1 Q_{0n} \end{array} \right.$$

$$\Rightarrow D_0 = Q_{0n}(C_0 \odot C_1) + \bar{Q}_{0n}(C_0 \oplus C_1)$$

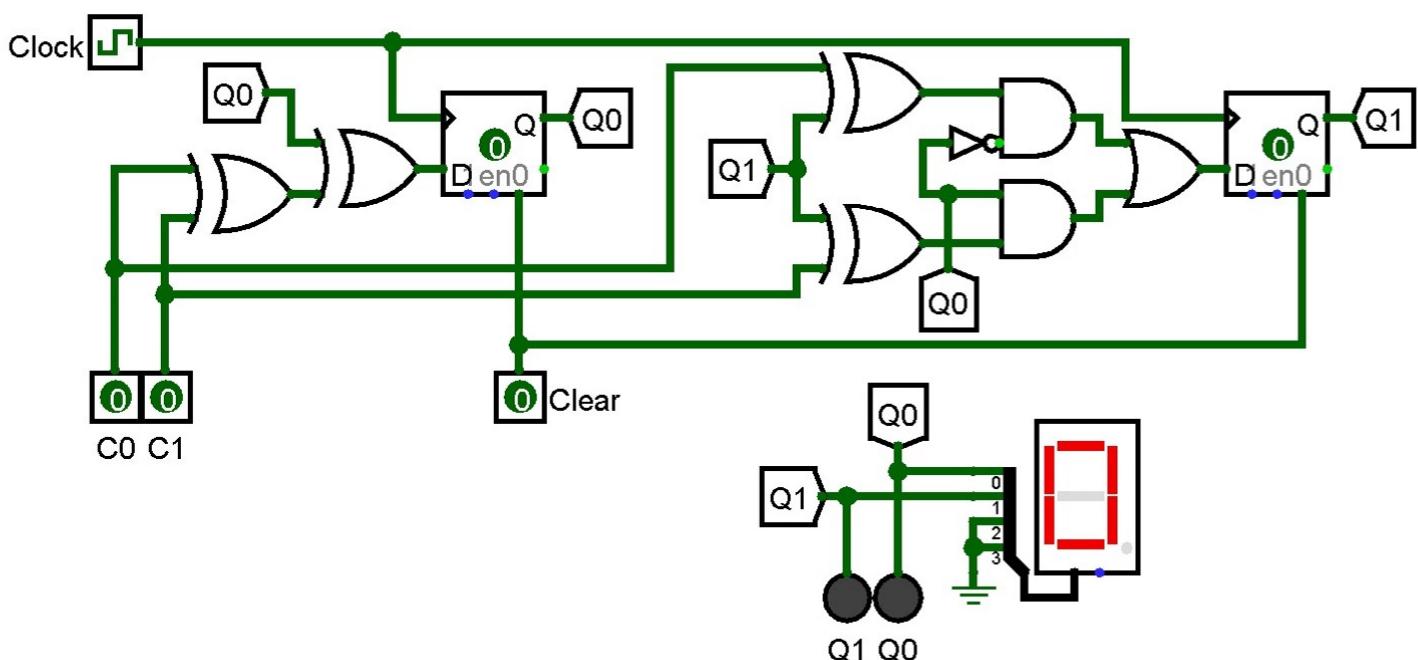
$$\Rightarrow D_0 = Q_{0n} \oplus (C_0 \oplus C_1)$$



$$D_1 = \bar{Q}_{1n} \bar{Q}_{0n} C_D + \bar{Q}_{1n} \bar{Q}_{0n} C_1 \\ + Q_{1n} \bar{Q}_{0n} \bar{C}_0 + Q_{1n} \bar{Q}_{0n} \bar{C}_1$$

$$\Rightarrow D_1 = \bar{Q}_{0n} (C_0 \oplus Q_{1n}) + Q_{0n} (C_1 \oplus Q_{1n})$$

## Logic Diagram:



III) Implementation using JK flipflop:

Circuit Excitation Table

$C_0$	$C_1$	$Q_{1in}$	$Q_{0in}$	$Q_{1out}$	$Q_{0out}$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	0	0	X	0	X
0	0	0	1	0	1	0	X	X	0
0	0	1	0	1	0	X	0	0	X
0	0	1	1	1	1	X	0	X	0
0	1	0	0	0	1	0	X	1	X
0	1	0	1	1	0	1	X	X	1
0	1	1	0	1	1	X	0	1	X
0	1	1	1	0	0	X	1	X	1
1	0	0	0	1	0	1	X	1	X
1	0	0	1	0	0	0	X	X	1
1	0	1	0	0	1	X	1	1	X
1	0	1	1	1	0	X	0	X	1
1	1	0	0	1	0	1	X	0	X
1	1	0	1	1	1	1	X	X	0
1	1	1	0	0	0	X	1	0	X
1	1	1	1	0	1	X	1	X	0

K-Map for  $J_1, K_1, J_0, K_0$

		Q <sub>1</sub> Q <sub>0</sub>	00	01	11	10
		C <sub>0</sub> C <sub>1</sub>	00	X		X
		01	(X)	1	1	X
		11	X		X	
		10	X	1	1	X

$$K_0 = C_0 \oplus C_1$$

		Q <sub>1</sub> Q <sub>0</sub>	00	01	11	10
		C <sub>0</sub> C <sub>1</sub>	00	X	X	
		01	1	X	X	1
		11		X	X	
		10	1	X	X	1

$$J_0 = C_0 \oplus C_1$$

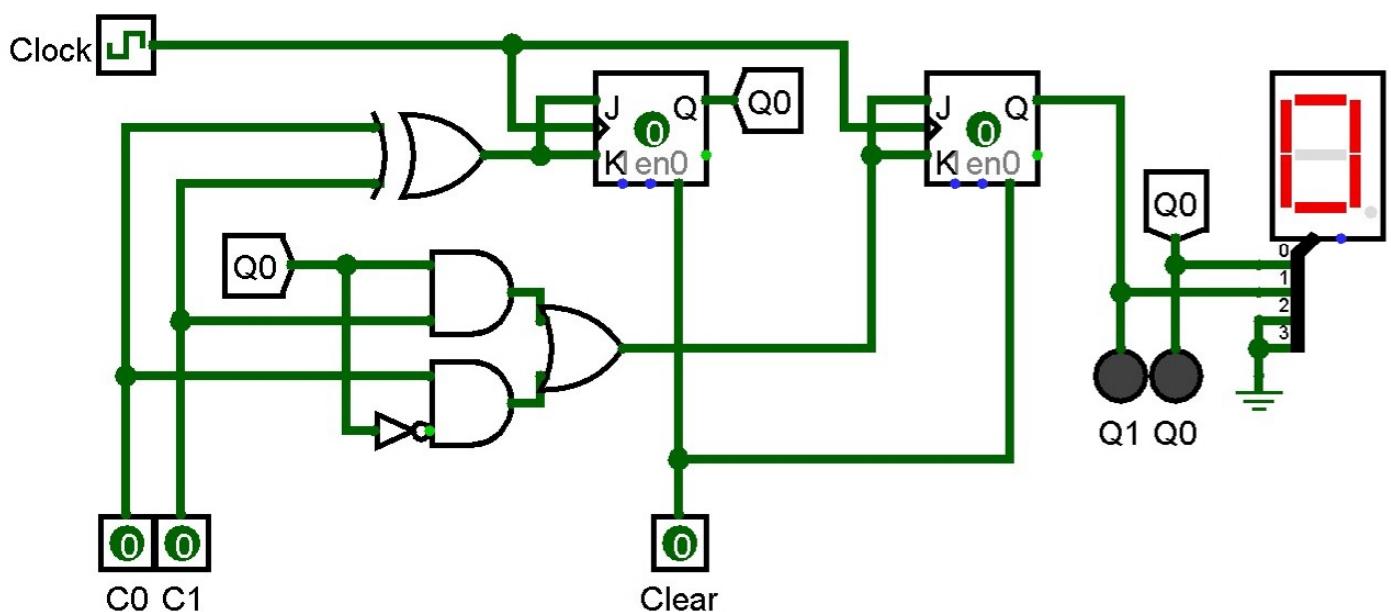
		Q <sub>1</sub> Q <sub>0</sub>	00	01	11	10
		C <sub>0</sub> C <sub>1</sub>	00	X	X	
		01	X	(X)	1	
		11	X	X	1	1
		10	X	X		1

$$K_1 = C_0 \bar{Q}_{0n} + C_1 Q_{0n}$$

		Q <sub>1</sub> Q <sub>0</sub>	00	01	11	10
		C <sub>0</sub> C <sub>1</sub>	00	X	X	
		01	1	X	X	
		11	1	X	X	
		10	1	X	X	

$$J_1 = C_0 \bar{Q}_{0n} + C_1 Q_{0n}$$

Logic Diagram:

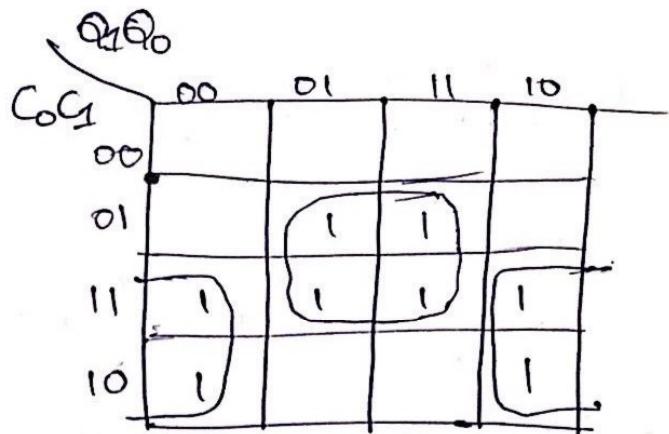
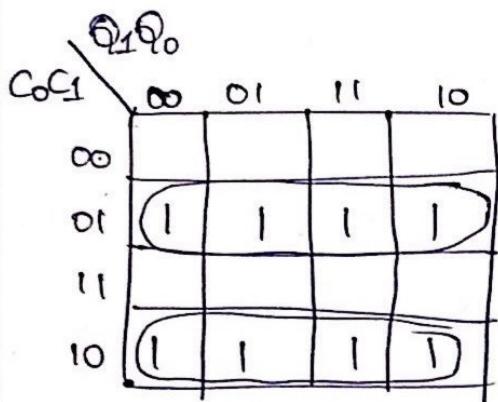


#### IV) Implementation of T flipflop

Circuit Excitation Table

$C_0$	$C_1$	$Q_{in}$	$Q_{out}$	$Q_{1\text{ nti}}$	$Q_{0\text{ nti}}$	$T1$	$T0$
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	1	0	1	1
0	1	1	0	1	1	0	1
0	1	1	1	0	0	1	1
1	0	0	0	1	1	1	1
1	0	0	1	0	0	0	1
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	1
1	1	0	0	1	0	1	0
1	1	0	1	1	1	1	0
1	1	1	0	0	0	1	0
1	1	1	1	0	1	1	0

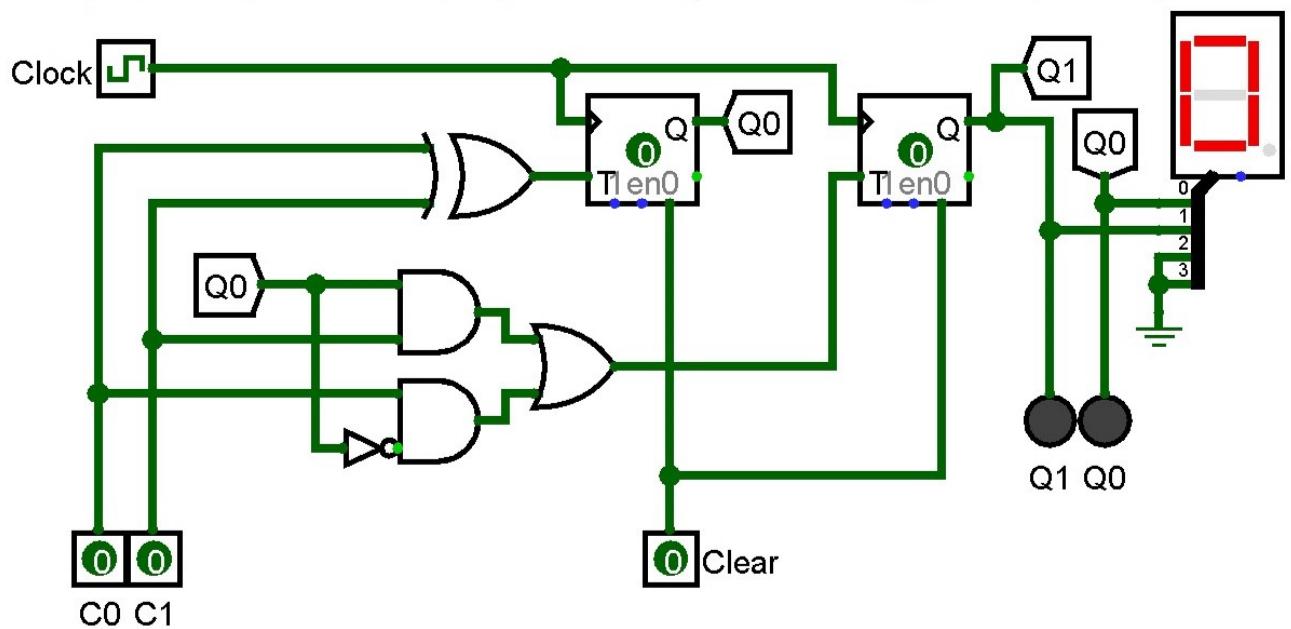
K-Map for T1, T0 :



$$T_0 = C_0 \oplus C_1$$

$$T_1 = C_1 Q_{on} + C_0 \bar{Q}_{on}$$

Logic Diagram:



Ans2:

Components we have,

D-flipflop, Shift Registers, Full Adder(1-bit, only one)  
(8-bit) (3)

- As we can use only a single full adder, we have to add the operands, bit by bit. We have to store them in shift registers.
- For each cycle of the clock, the full adder should add each bit of the operands along with carry obtained by addition of previous bits (i.e., previous clock cycle)
- To store the carry obtained from previous clock cycle we can use a D-flip flop.
- To store the sum obtained in each cycle, we can use another shift register. After 8 more clock cycles we get our desired outputs, the sum of operands and final carry.

Procedure to add numbers:

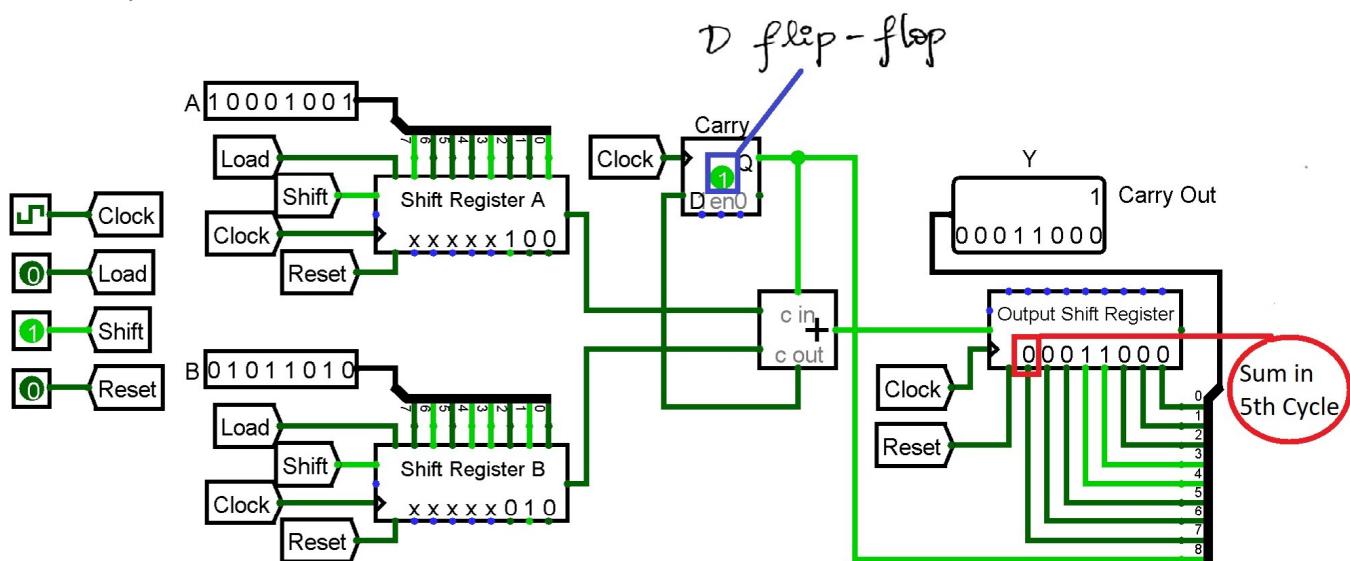
- ① Set Load = 1, then toggle clock once. Inputs are loaded into input shift registers.
- ② Set Load = 0, Shift = 1. Toggle clock once more
- ③ Toggle clock for 8 times. The sum of operands is stored in the shift register output
- ④ Set Reset = 1 to reset the input shift registers.

The state of the circuit is shown after 5<sup>th</sup> clock cycle, for the inputs  $A = 10001001$  and  $B = 01011010$ .

After the 4<sup>th</sup> clock cycle, the sum of bits from A and B is 0, which is stored in the shift register as MSB. And the carry 1 is stored in the Dflip-flop.

So, during 5<sup>th</sup> cycle, the sum is  $\underline{1} + \underline{0} + \underline{1} = \underline{0}$  is stored as the new MSB, and the carry 1 is stored in the D flip-flop.

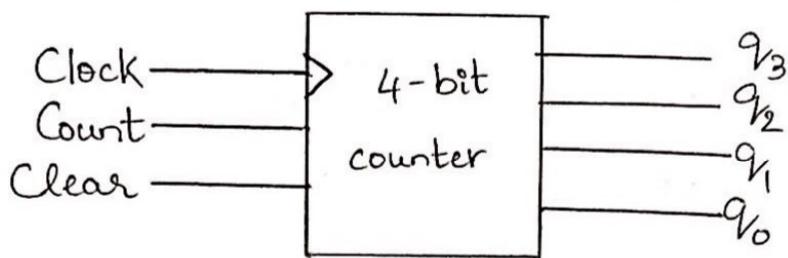
After 8 cycles, all bits of the sum are in the right places. The input shift registers have to be reset for any further use.



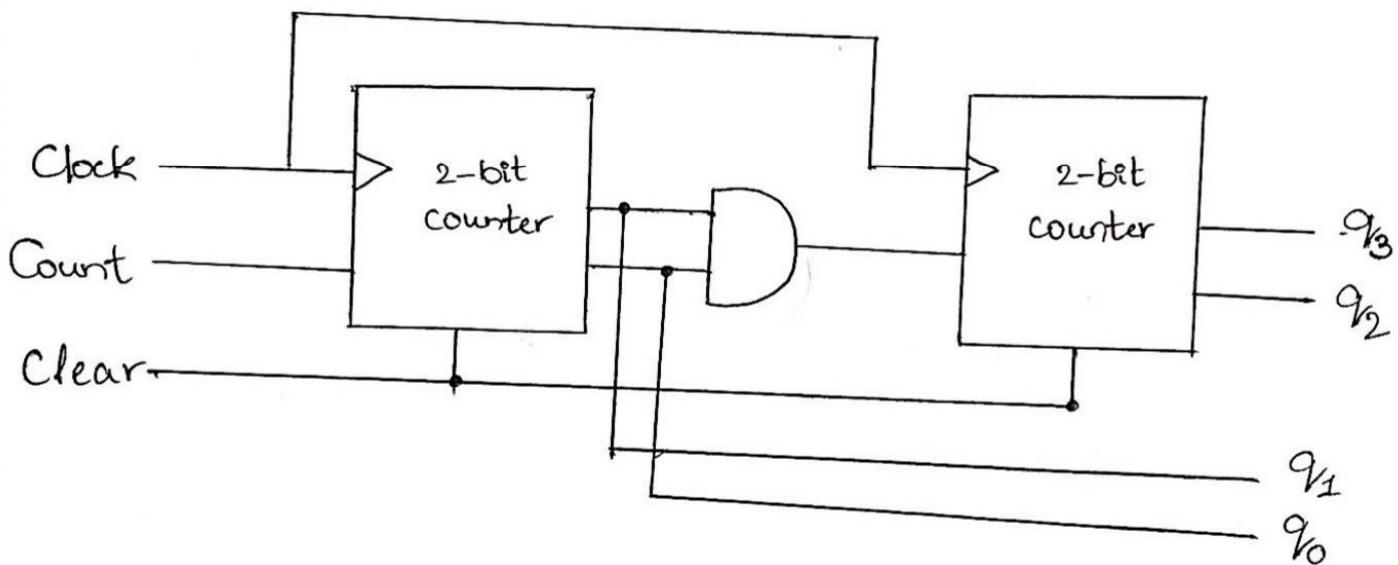
Ans 3

Block diagram of 4-bit counter:

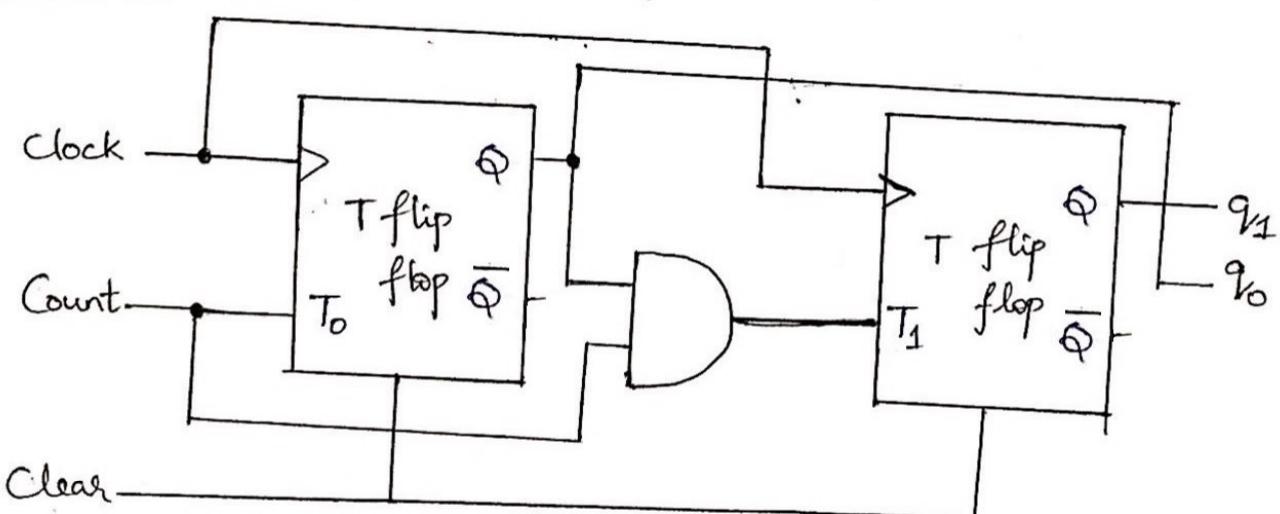
Count: when 1 increments  
when 0 stops counting



Implementation of 4-bit counter using 2-bit counter:



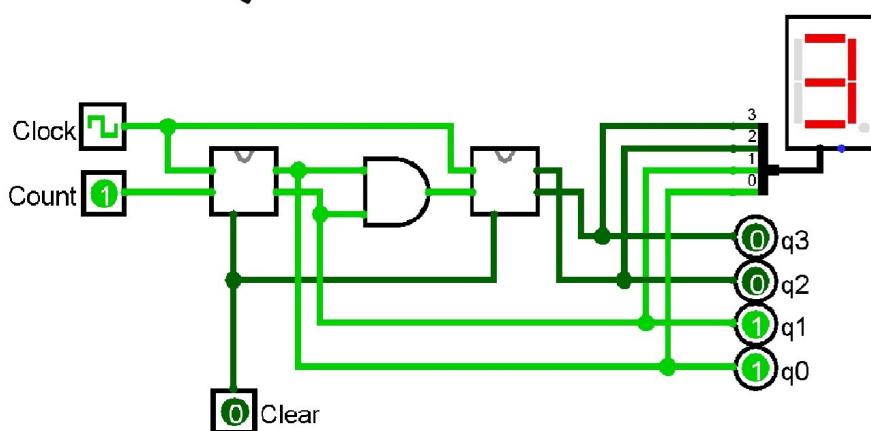
2-bit counter (implemented using T flip flops):



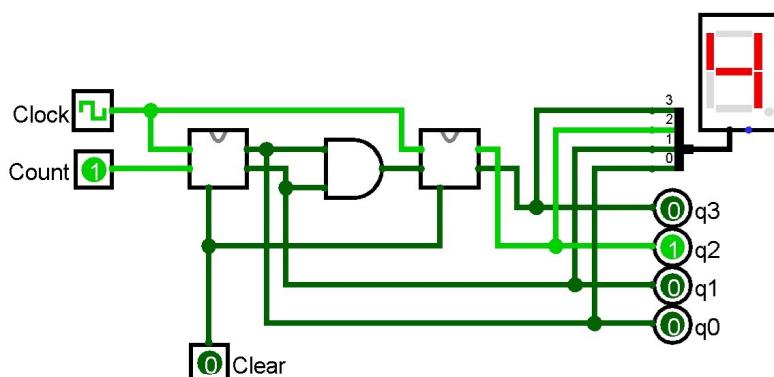
Right side 2-bit counter should increment only when the left side counter reaches 3(BIN 11), i.e.,  $q_1 = 1$  and  $q_0 = 1$ .

Hence feed  $\underline{(q_1 \cdot q_0)}$  as Count for the right side counter  
 $q_1 \text{ AND } q_0$

After 3 clock cycles,



After 4 clock cycles,



Notice that after 3 clock cycles,  $q_1 = 1$  and  $q_0 = 0$ . Hence, the value in the right counter was incremented i.e.,  $q_2 = 0$ , before, but now,  $q_2 = 1$ , and  $(q_3, q_2)$  represent DEC 1, which is the value in the right counter.