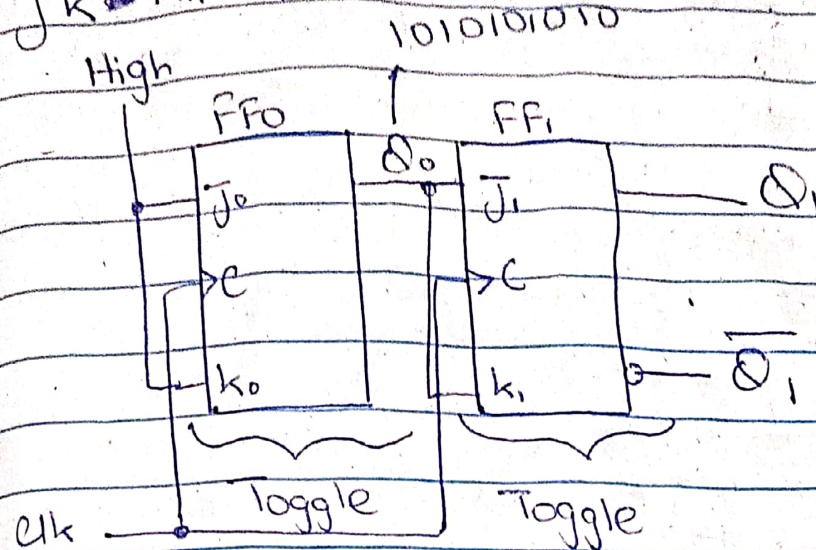
Chap 9

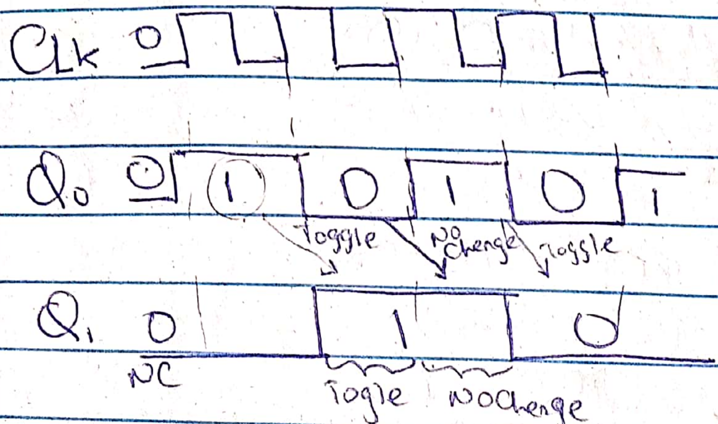
Synchronous Counters:-

JK flipflop:-



Q_1 is dependent on(Q_0).

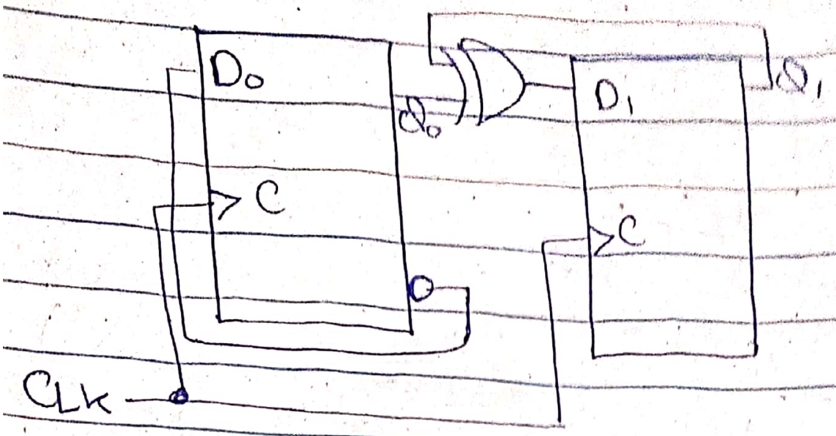
Toggle
No change
Toggle

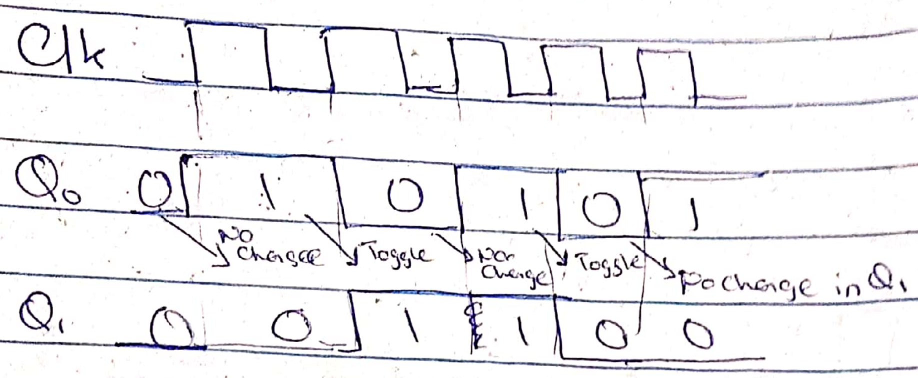


Since Q_0 is passed in ff1 So Q_0 is Toggl Throughout and because of that FF1 be output Q_1 become Toggle then NO Change then toggle.

When Q_0 is 1, Q_1 is toggle in next clock trigger and when Q_0 is 0, Q_1 remains same (No Change) in next clock pulse

D flip-flop:-





3 bit synchronous Counter:-

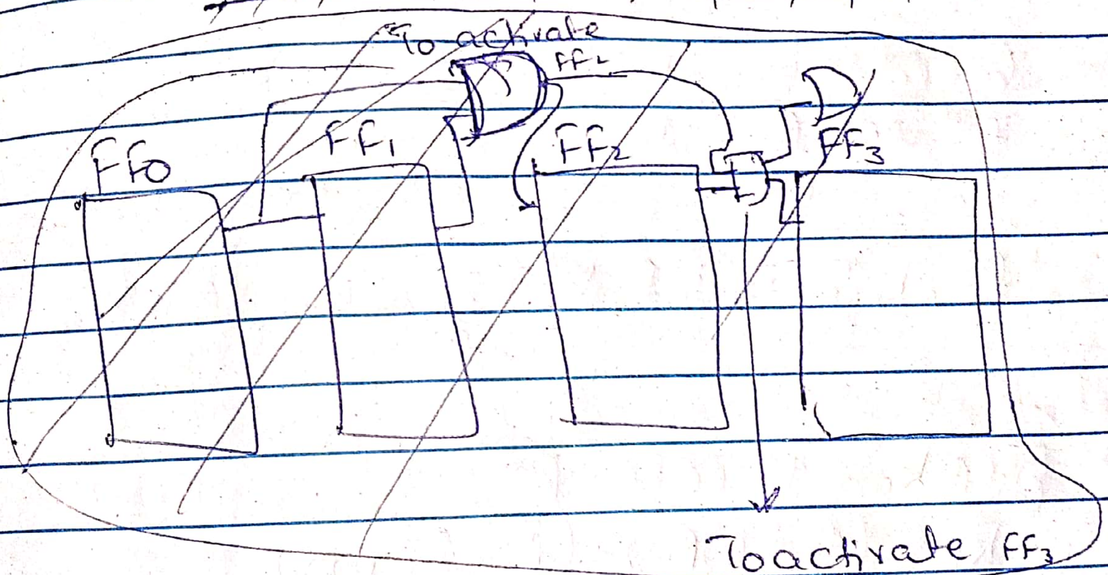
Clock pulse	Q_2	Q_1	Q_0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	$\boxed{1}$	$\boxed{1}$
4	1	0	0
5	1	0	1
6	1	1	0
7		$\boxed{1}$	$\boxed{1}$
8 (recycles)	0	0	0

When Q_0 and Q_1 are 1 Q_2 Toggle

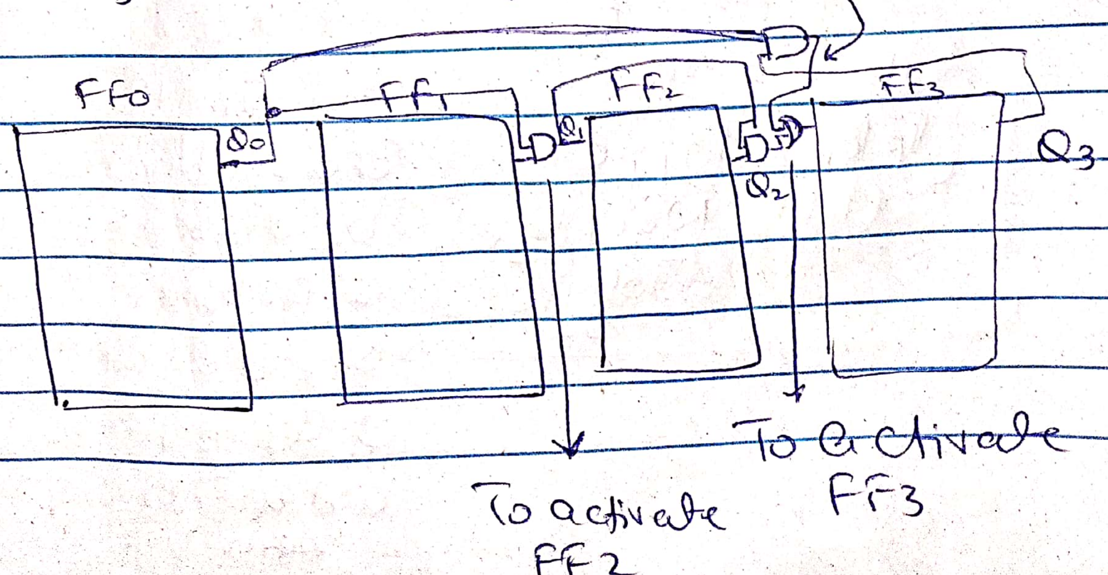
For 4 bit Synchronous Counter :-

Q_3, Q_2, Q_1, Q_0

(So Q_3 changes state when $Q_2 Q_1 Q_0$ are 1) and (Q_2 changes state when $Q_1 Q_0$ are 1) and (Q_1 changes state when Q_0 is 1)



$J_3 = Q_0 Q_1 Q_2 + Q_0 Q_3$



Up-Down Sequence for a 3 bit Counter:-

Clock pulse	Up	Q_2	Q_1	Q_0	Down
0	↓	0	0	0	↑
1	↓	0	0	1	↑
2	↓	0	1	0	↑
3	↓	0	1	1	↑
4	↓	1	0	0	↑
5	↓	1	0	1	↑
6	↓	1	1	0	↑
7	↓	1	1	1	↑

1

$$\underbrace{0, 1, 2, 3, 4, 5}_{\text{UP}}, \underbrace{4, 3, 2}_{\text{Down}}, \underbrace{3, 4, 5, 6, 7}_{\text{UP}}, 6, \text{etc}$$

$$J_0 = k_0 = 1$$

$$J_1 = k_1 = (Q_0 . UP) + (\bar{Q}_0 . Down)$$

$$J_2 = k_2 = (Q_0 . Q_1 . UP) + (\bar{Q}_0 . \bar{Q}_1 . Down)$$

$$\boxed{\begin{array}{l} J_1 = k_1 = \text{Toggle} \\ \qquad\qquad \text{No change} \\ \qquad\qquad \text{Toggle} \end{array}}$$

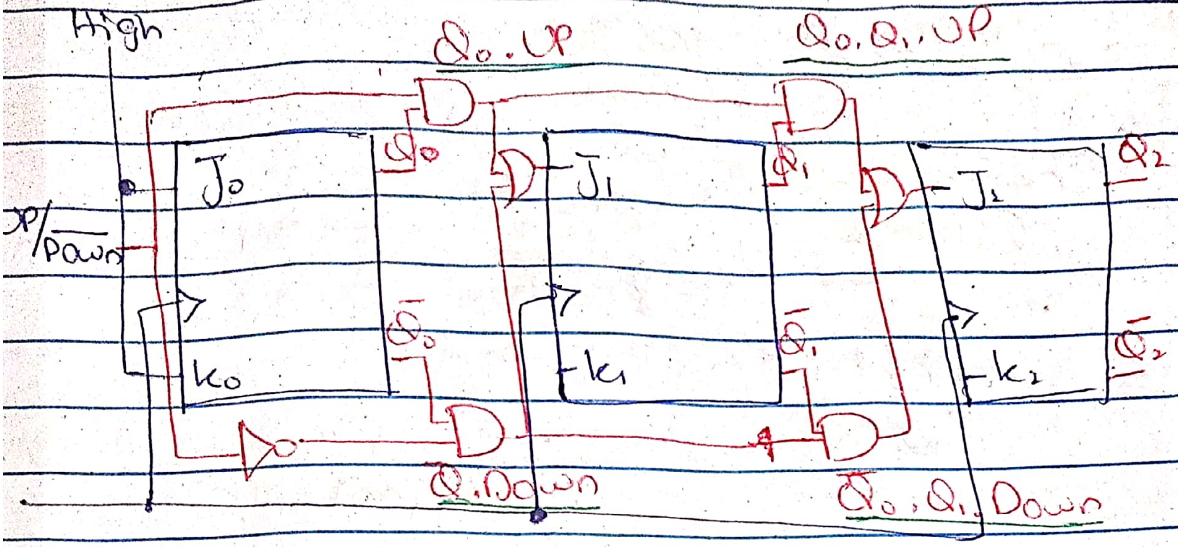
when Q_1 and Q_0 are 1 Q_2 goes up and becomes 1. And when Q_1 and Q_0 are 0, Q_2 goes down and becomes zero

Since:-

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = (Q_0 . UP) + (\overline{Q_0} . Down)$$

$$J_2 = K_2 = (Q_0 . Q_1 . UP) + (\overline{Q_0} . \overline{Q_1} . Down)$$



Decoding:-

For decoding the three FFs will be & outputs (Q) will be joined to AND in decimal combination like

To decode 6

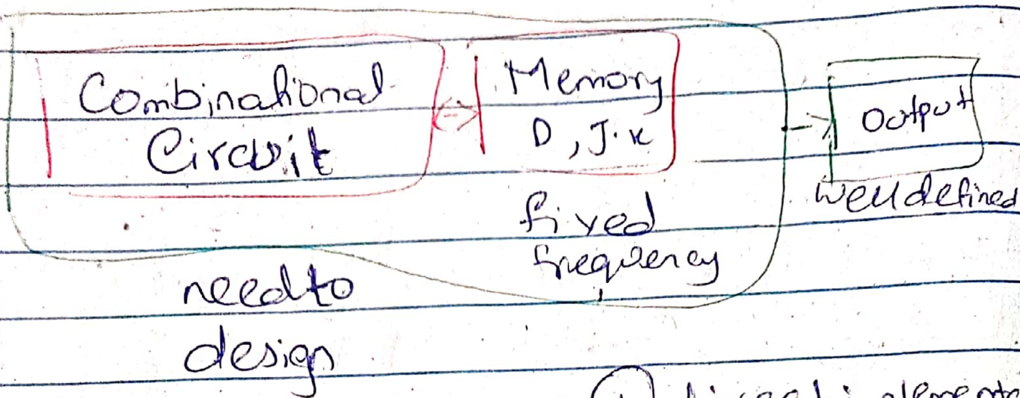
$Q_2 \; Q_1 \; \overline{Q_0}$ must be AND

$1 \; 1 \; 0 = 6$

Designing of Synchronous Counters:-

for memory we use either J.K flipflop or D-flip-flop.

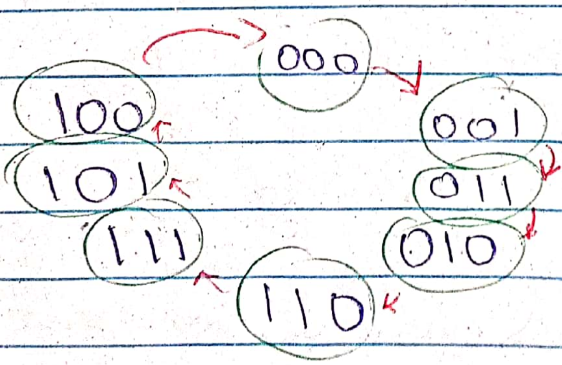
Finite State Machine



1. direct implementatio
2. Truth Table
3. k-Maps

6 Steps To Create Asynchronous Counte

Step 1:- State Diagram



For a 3 bit Grey Code Coun

Step 2 : Next State Table

Present States			Next States		
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Step 3 :- Flip Flop Transition Table

Transition table ~~fa~~ for a J-k ff :

Output Transitions			Flip-Flop Inputs	
Q_N		Q_{N+1}	J	k
0	$\rightarrow$	0	0	X
0	$\rightarrow$	1	1	X
1	$\rightarrow$	0	X	1
1	$\rightarrow$	1	X	0

Q_N : Present States

Q_{N+1} : next state

X : "don't care"

Scanned with CamScanner

J	K	CLK	Q	$\bar{Q}$	
0	0	↑	Q	$\bar{Q}$	No Change
0	1	↑	0	1	Reset
1	0	↑	1	0	Set
1	1	↑	$\bar{Q}_0$	Q_0	Toggle

For D:- Flip-flop step 3

Output lenght		Inputs D ~~Inputs~~
Q_N	Q_{N+1}	
0 →	0	0
0 →	1	1
1 →	0	0
1 →	1	1

→ inputs for 'D'

D	CLK	Q	
0	↑	0	Reset
1	↑	1	Set

Scanned with CamScanner

Step 4: Karnaugh Maps

Transition Table

Output Transition		Flip-Flop Inputs	
Q_N	Q_{N+1}	J	K
0 →	0	0	X
0 →	1	1	X
1 →	0	X	1
1 →	1	X	0

Next Table

Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

The karnaugh maps for (J_0, K_0), (J_1, K_1), (J_2, K_2) will be mapped differently (i.e different for J_0 and K_0). The states will be seen from the transition table and then the corresponding J and K will be plotted on karnaugh map according to there Q_2, Q_1, Q_0.

like Q_0 was initially 0 and at Next State it became 1. So according To transition Table when Q goes from 0 → 1, J = 1 and K = X. and so accordingly plot for Q_1 by looking at Q_1 Next State and Same for Q_2

Plot these by yourself for Practice

$Q_2Q_1 \backslash Q_0$	0	1
00	1	X
01	0	X
11	1	X
10	0	X

$\rightarrow \bar{Q}_2\bar{Q}_1$

$\rightarrow Q_2Q_1$

J_0 Map

$Q_2Q_1 \backslash Q_0$	0	1
00	X	0
01	X	1
11	X	
10	X	1

$\rightarrow \bar{Q}_2 Q_1$

$\rightarrow Q_2\bar{Q}_1$

K_0 Map

$Q_2Q_1 \backslash Q_0$	0	1
00	0	1
01	X	X
11	X	X
10	X	X

$\rightarrow \bar{Q}_2 Q_0$

J_1 Map

$Q_2Q_1 \backslash Q_0$	0	1
00	X	X
01	0	0
11	0	1
10	X	X

$\rightarrow Q_2Q_0$

K_1 Map

$Q_2Q_1 \backslash Q_0$	0	1
00	0	0
01	1	0
11	X	X
10	X	X

$\rightarrow Q_1\bar{Q}_0$

J_2 Map

$Q_2Q_1 \backslash Q_0$	0	1
00	X	X
01	X	X
11	0	0
10	1	0

K_2 Map

$\bar{Q}_1\bar{Q}_0$

Step 5:- logic expression for flip-flop inputs

$$J_0 = Q_2 Q_1 + \overline{Q}_2 \overline{Q}_1 = \overline{Q_2 \oplus Q_1}$$

$$\cancel{J_1 = k_0}$$

$$K_0 = Q_2 \overline{Q}_1 + \overline{Q}_2 Q_1 = Q_2 \oplus Q_1$$

$$J_1 = \overline{Q}_2 Q_0$$

$$K_1 = Q_2 Q_0$$

$$J_2 = Q_1 \overline{Q}_0$$

$$K_2 = \overline{Q}_1 \overline{Q}_0$$

Step 6. Counter Implementation