

Counters :-

→ It is an sequential circuit capable of counting the number of clock pulses arriving at its clock input.

→ There are 2 types of counters,

* Ripple / Asynchronous / Non-Synchronous / serial counters. UP DOWN

* Synchronous / parallel counter

Synchronous Counters.	Asynchronous Counters.
(i) All the flip flops are clocked simultaneously.	(i) All the flip flops are not clocked simultaneously, the clock pulse is given only to the first flip flop.
(ii) Only first flop inputs are given to "1".	(ii) All the flip flops inputs are given to "1".
(iii) propagation delay is less.	(iii) propagation delay is more.
(iv) Circuit diagram design is complicated.	(iv) Circuit design is simpler.

General Procedure:-

* Truth Table

* Logic Circuit

* Timing Diagram

} → steps to be followed in all design.

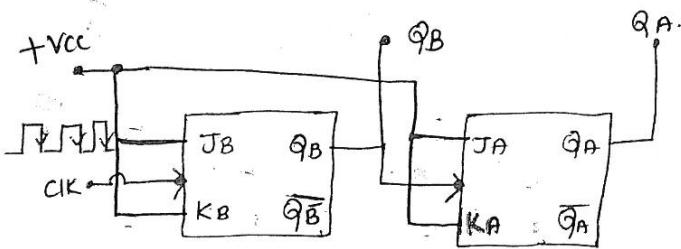
} → c check whether the clock pulse is +ve/-ve edge triggering

1). Design 2bit / Mod-4 Asynchronous UP counter.

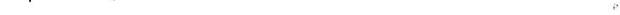
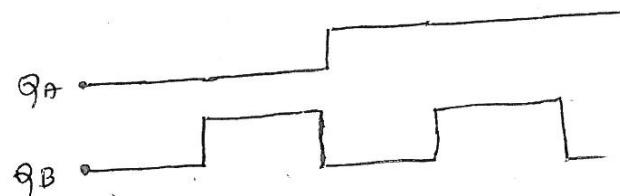
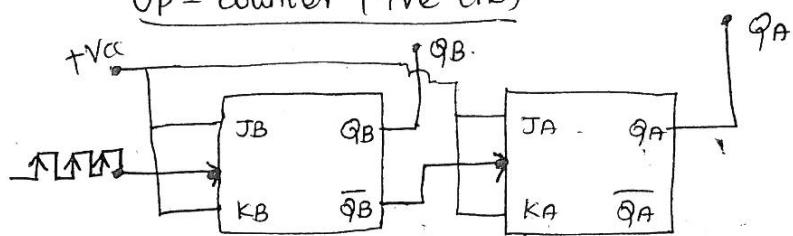
→ To design 2 bit counter, we require 2 flip-flops.

CLK.	UP	
	QA (MSB)	QB (LSB)
1	0	0
2	0	1
3	1	0
4	1	1
5	0	0
6	0	1

UP - Counter (-Ve CLK).



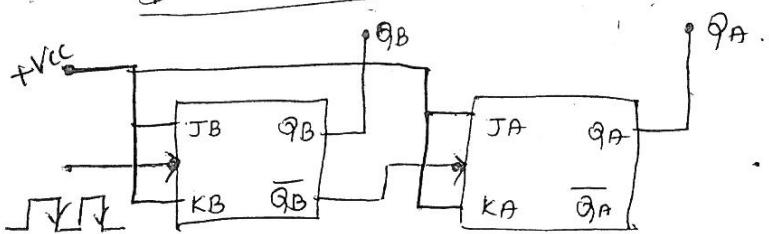
UP - Counter (+Ve CLK)



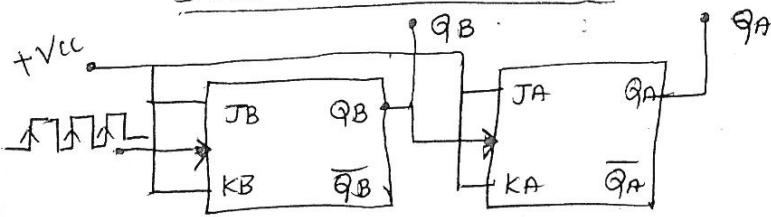
For Down Counter:

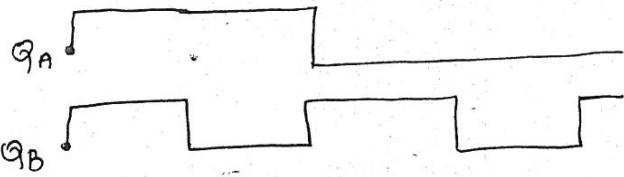
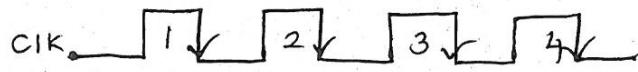
CLK.	Down	
	QA (MSB)	QB (LSB)
1	1	1
2	1	0
3	0	1
4	0	0
5	1	1
6	1	0

Down counter (-Ve CLK).



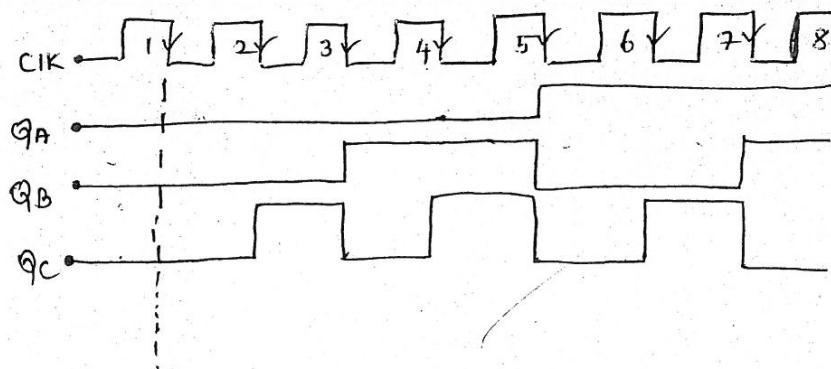
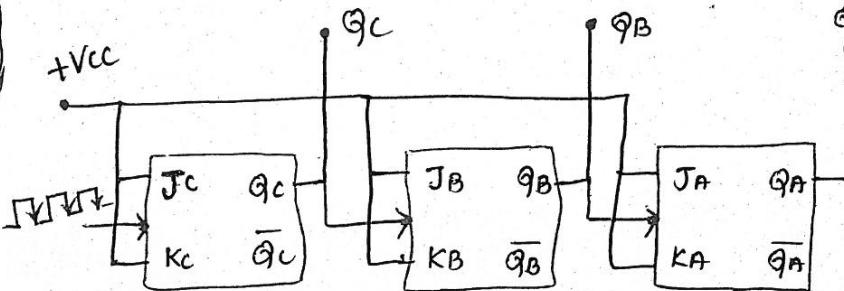
Down counter (+Ve CLK).





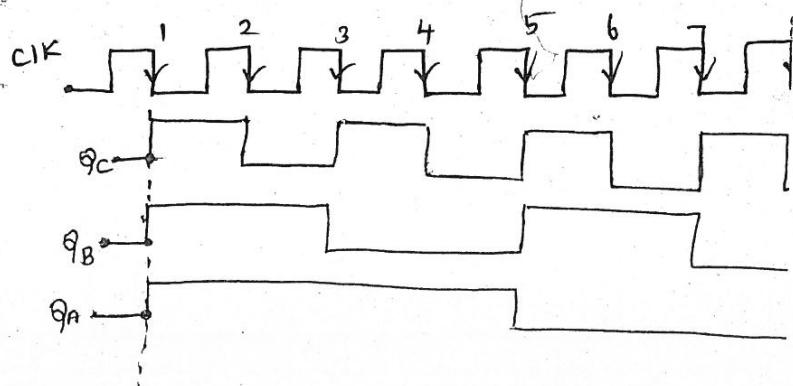
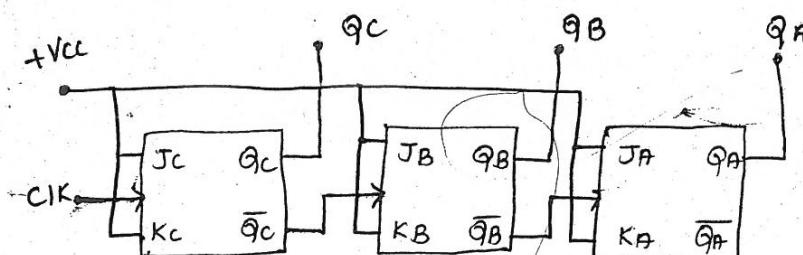
(2). Design a 3-bit Asynchronous UP Counter / (MOD-8)

CLK.	Q _A	Q _B	Q _C
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1
9	0	0	0

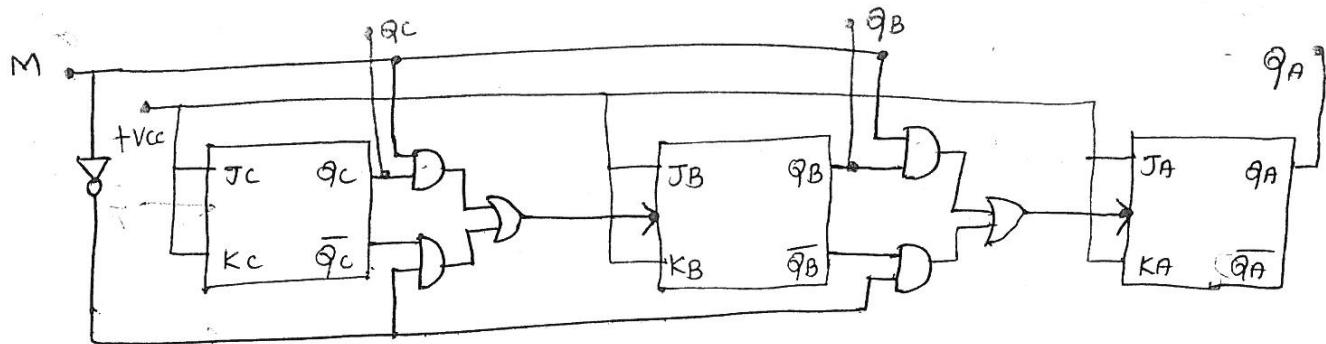


For Down counter,

CLK	Q _A	Q _B	Q _C
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0
9	1	1	1



3) 3 bit Asynchronous UP / Down Counter:-

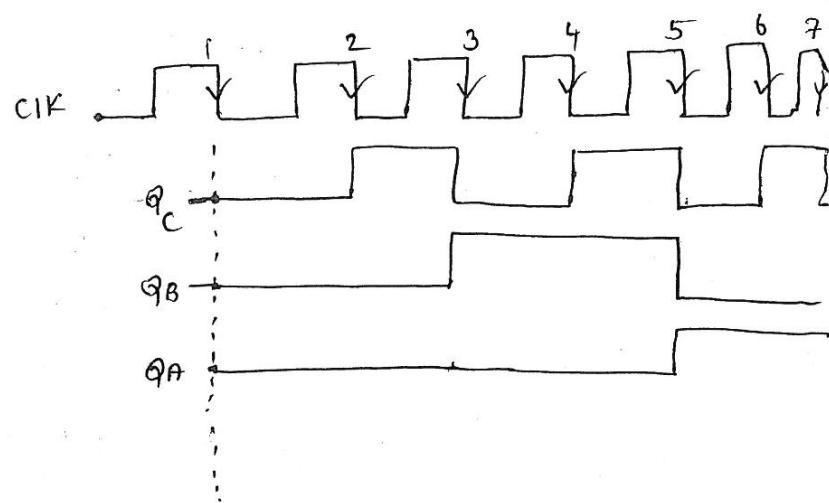
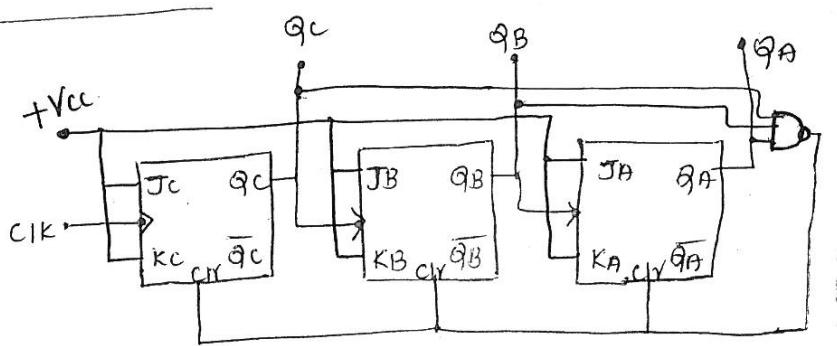


$M = 1 \rightarrow$ UP counter

$M = 0 \rightarrow$ Down counter.

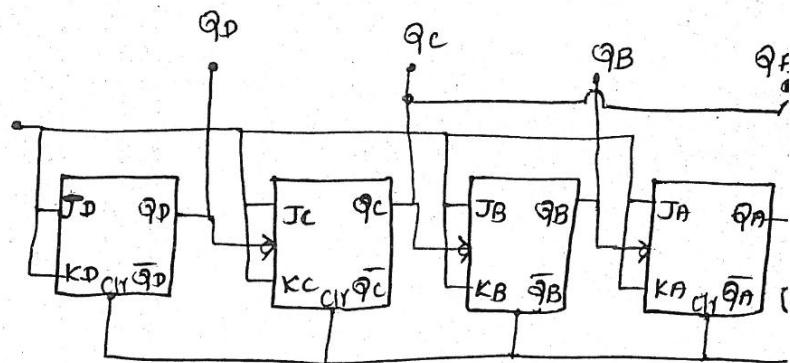
4) Design mod-7 Asynchronous Counter.

CLK.	Q_A	Q_B	Q_C .
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0

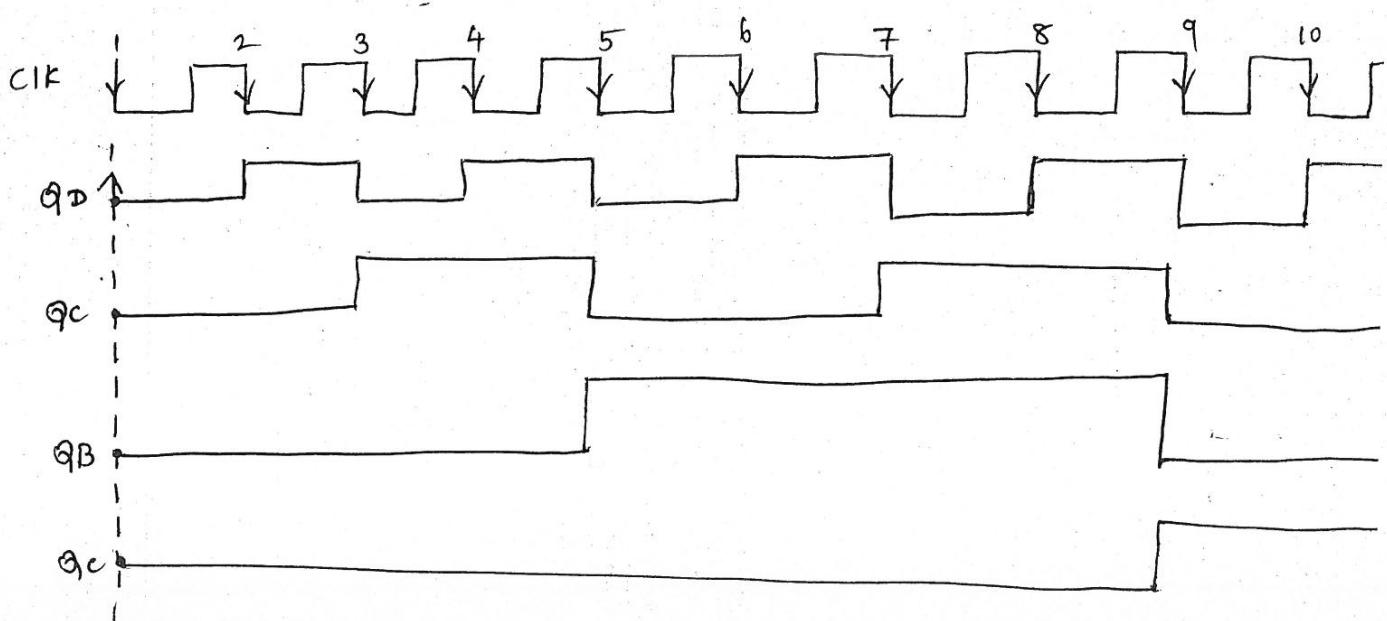


5) Design mod-10 Asynchronous - UP Counter.

CLK	QA	QB	QC	QD
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1

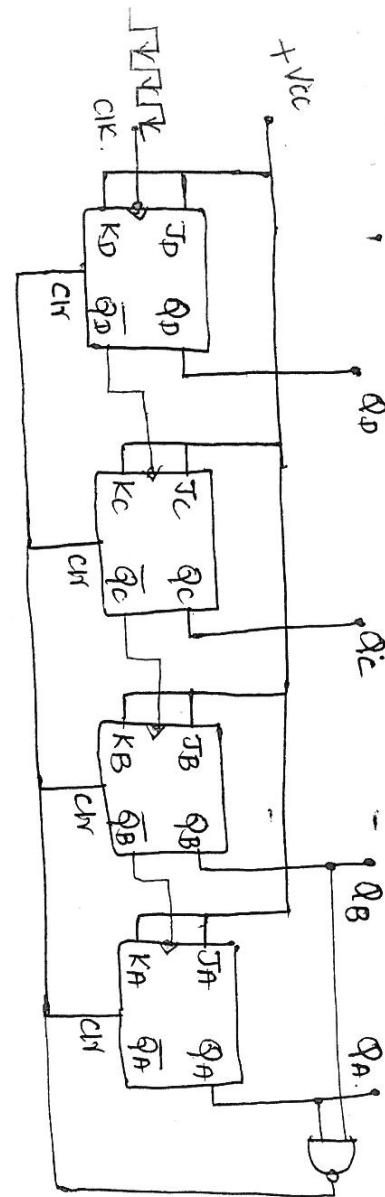


$$\begin{array}{cccc}
 1 & 0 & 1 & 0 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 0 & 0 & 0 & 0
 \end{array}$$

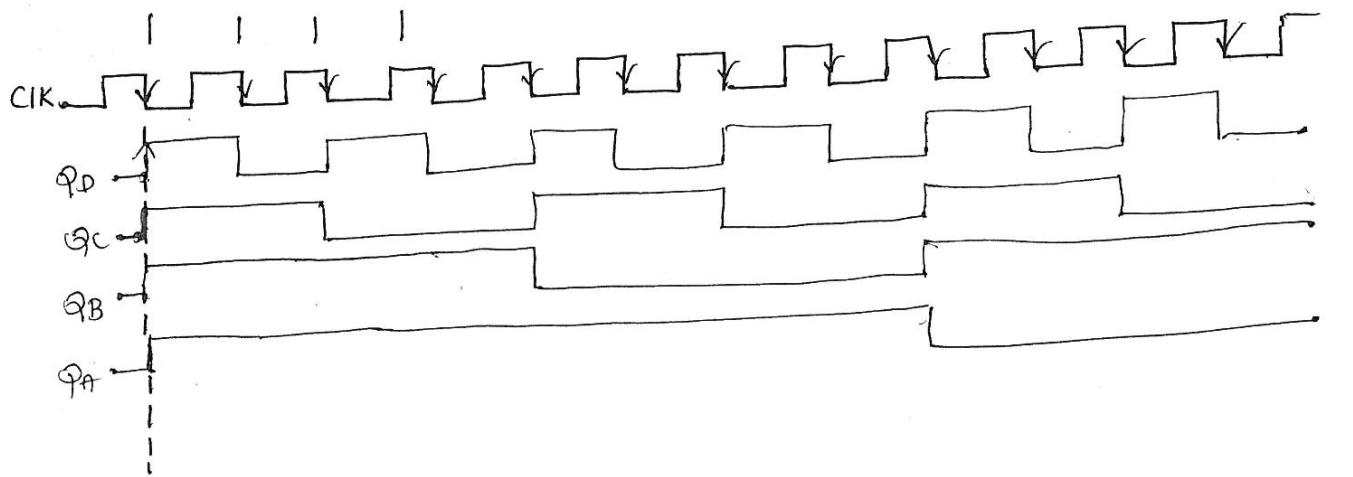


1) Design Mod-12 Asynchronous Down Counter.

CLK	QA	QB	QC	QD
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	0	1	1	1
10	0	1	1	0
11	0	1	0	1
12	0	1	0	0



13. 0 0 1 1
 ↓ ↓ ↓ ↓
 1 1 1 1



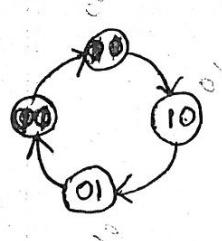
Synchronous counter.

- When counter is clocked such that each flip-flop in the counter is triggered at the same time, the counter is called Synchronous counter.
- clock signal is connected in parallel to clock inputs of both the flip-flops. But Q_A output of the first stage is used to drive the J and K inputs of the second stage.

1) Design a 2bit-Synchronous UP counter and Down Counter.

2bit UP Counter:

State Diagram.



Truth Table.

CIK	QA	QB	QA+1	QB+1	JA KA	JB
1	0	0	0	1	0 X	1
2	0	1	1	0	1 X	X
3	1	0	1	1	X 0	1
4	1	1	0	0	X 1	X

JA	QA	QB
0	0	1
X	X	X

$JA = QB$

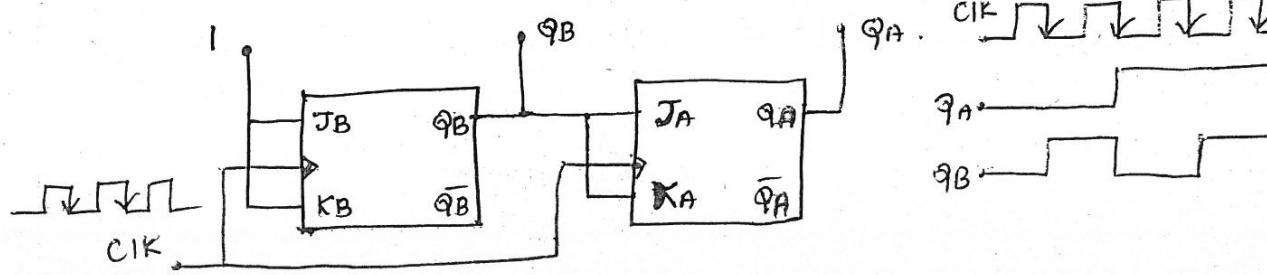
JK _A	QA	QB
X	0	X
0	1	1

$KA = Q_B$

JB	QA	QB
i	X	
1	X	

KB	QA	QB
X	1	
X	1	

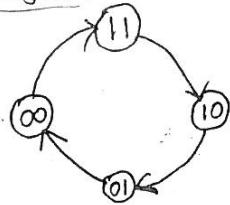
$$JB = 1 \quad KB = 1$$



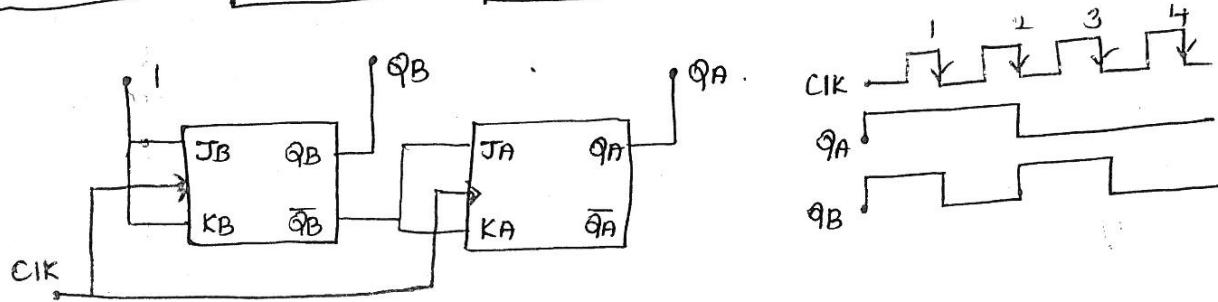
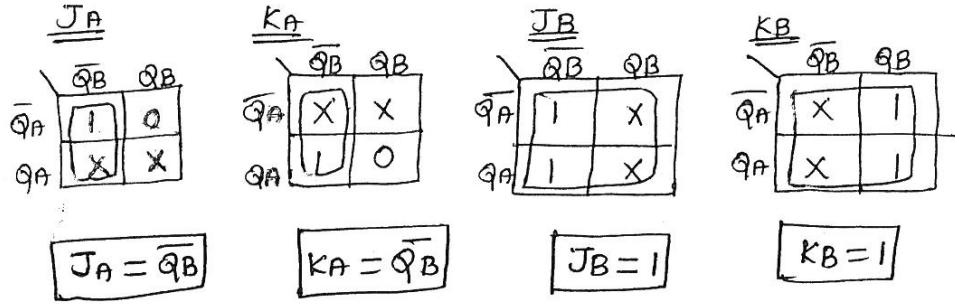
2-bit Down Counter:-

Truth Table.

State Diagram



CIK	QA	QB	QA+1	QB+1	JA	KA	JB	KB
1	1	1	1	0	X	0	X	1
2	1	0	0	1	X	1	1	X
3	0	1	0	0	0	X	X	1
4	0	0	1	1	1	X	1	X

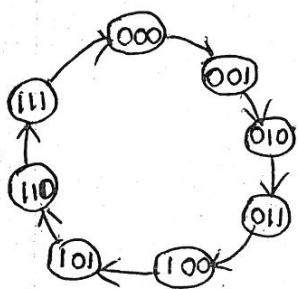


) Design of 3-bit Synchronous Up-Counter using JK - Flip-flop.

Truth table.

CIK	QA	QB	QC	QA+1	QB+1	QC+1	JA	KA	JB	KB	Jc	Kc
1	0	0	0	0	0	1	0	X	0	X	1	X
2	0	0	1	0	1	0	0	X	1	X	X	1
3	0	1	0	0	1	1	0	X	X	0	1	X
4	0	1	1	1	0	0	1	X	X	0	X	1
5	1	0	0	1	0	1	X	0	1	X	X	1
6	1	0	1	1	1	0	X	0	X	0	1	X
7	1	1	0	1	1	1	X	0	X	1	X	1
8	1	1	1	0	0	0	X	1	X	1	X	1

State Diagram



<u>J_A</u>				
$\bar{Q}_B\bar{Q}_C \bar{Q}_BQ_C Q_B\bar{Q}_C Q_BQ_C$				
\bar{Q}_A	0	0	1	0
Q_A	X	X	X	X
\bar{Q}_A	0	0	1	0

$J_A = \bar{Q}_B Q_C$

<u>K_A</u>				
$\bar{Q}_B\bar{Q}_C \bar{Q}_BQ_C Q_B\bar{Q}_C Q_BQ_C$				
\bar{Q}_A	X	X	X	X
Q_A	0	0	1	0
\bar{Q}_A	0	0	1	0

$K_A = \bar{Q}_B Q_C$

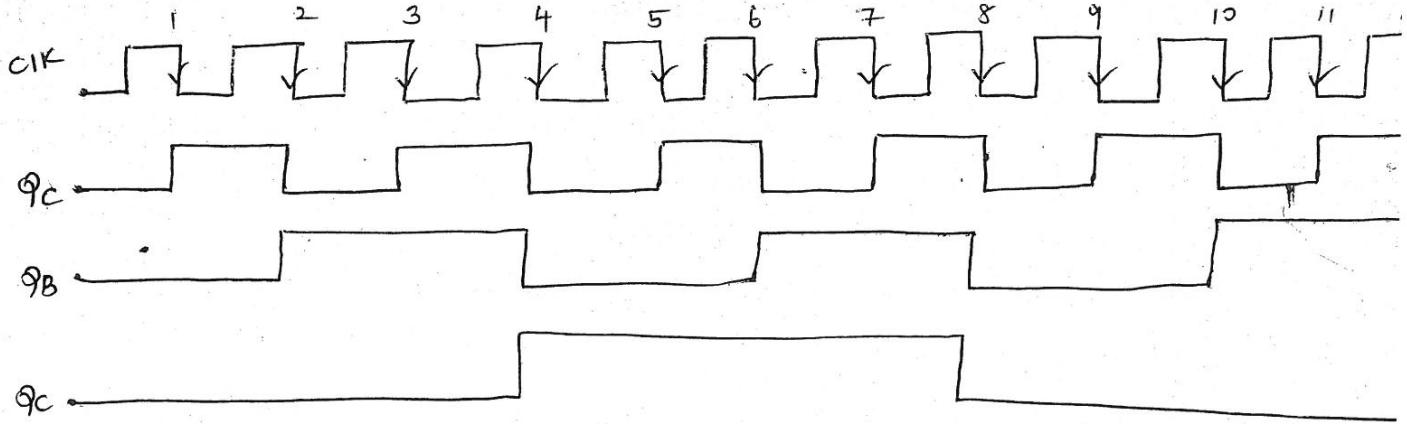
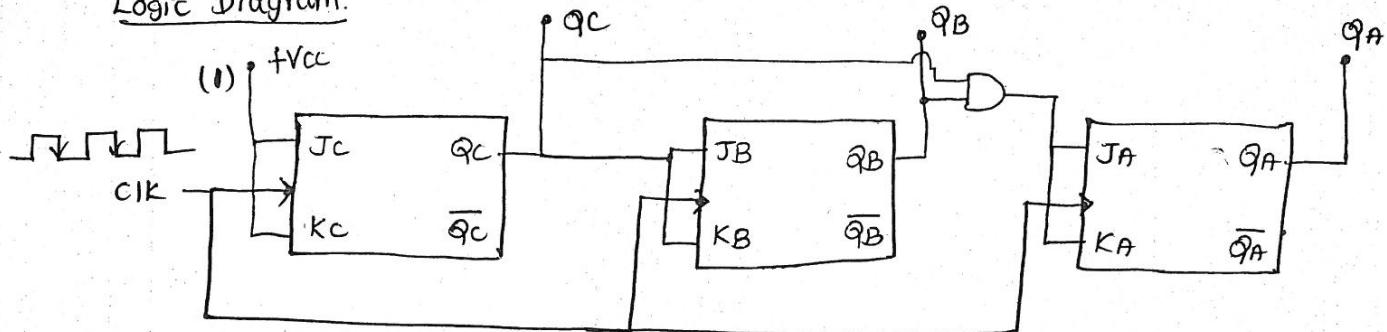
<u>J_B</u>				
$\bar{Q}_B\bar{Q}_C \bar{Q}_BQ_C Q_B\bar{Q}_C Q_BQ_C$				
\bar{Q}_A	0	1	X	X
Q_A	0	1	X	X

<u>K_B</u>				
$\bar{Q}_B\bar{Q}_C \bar{Q}_BQ_C Q_B\bar{Q}_C Q_BQ_C$				
\bar{Q}_A	X	X	1	0
Q_A	X	X	1	0

<u>J_C</u>				
$\bar{Q}_B\bar{Q}_C \bar{Q}_BQ_C Q_B\bar{Q}_C Q_BQ_C$				
\bar{Q}_A	1	X	1	X
Q_A	1	X	1	X

<u>K_C</u>				
$\bar{Q}_B\bar{Q}_C \bar{Q}_BQ_C Q_B\bar{Q}_C Q_BQ_C$				
\bar{Q}_A	X	1	X	1
Q_A	X	1	X	1

Logic Diagram:



1) A counter has 14 stable states 0000 through 1101. If the input frequency is 50 kHz, what will be its output frequency?

Soln:- $\frac{50\text{kHz}}{14} = 3.57 \text{ kHz}$

2) The t_{pd} for each FF is 50 ns, determine the maximum operating frequency for Mod-32 ripple counter.

Soln:- $f_{ax(\text{ripple})} = \frac{1}{5 \times 50\text{ns}} = 4 \text{ MHz}$

1) Design mod-7 counter. Using Synchronous.

Truth Table

QA	QB	QC	Q _{A+1}	Q _{B+1}	Q _{C+1}	J _A K _A	J _B K _B	J _C K _C
0	0	0	0	0	1	0 X	0 X	1 X
0	0	1	0	1	0	0 X	1 X	X 1
0	1	0	0	1	1	0 X	X 0	1 X
0	1	1	1	0	0	1 X	X 1	X 1
1	0	0	1	0	1	0 X	0 X	1 X
1	0	1	1	1	0	0 X	1 X	X 1
1	1	0	0	0	0	X 1	X 1	0 X
1	1	1	X	X	X	X X	X X	X X

K-map Simplification:-

		JA			
		Q _B Q _C	Q _B QC	QBQC	Q _B QC
		0	0	1	0
Q _A	Q _B	0	1	X	0
Q _A	Q _B	4	5	7	6

$$JA = Q_B Q_C$$

		KA			
		Q _B Q _C	Q _B QC	QBQC	Q _B QC
		0	1	3	2
Q _A	Q _B	X	X	X	X
Q _A	Q _B	4	5	7	6

$$KA = 1$$

		JB			
		Q _B Q _C	Q _B QC	QBQC	Q _B QC
		0	1	3	2
Q _A	Q _B	0	1	X	X
Q _A	Q _B	4	5	X	X

$$JB = Q_C$$

		KB			
		Q _B Q _C	Q _B QC	QBQC	Q _B QC
		0	1	3	2
Q _A	Q _B	X	X	1	0
Q _A	Q _B	4	5	X	1

$$KB = Q_A + Q_C$$

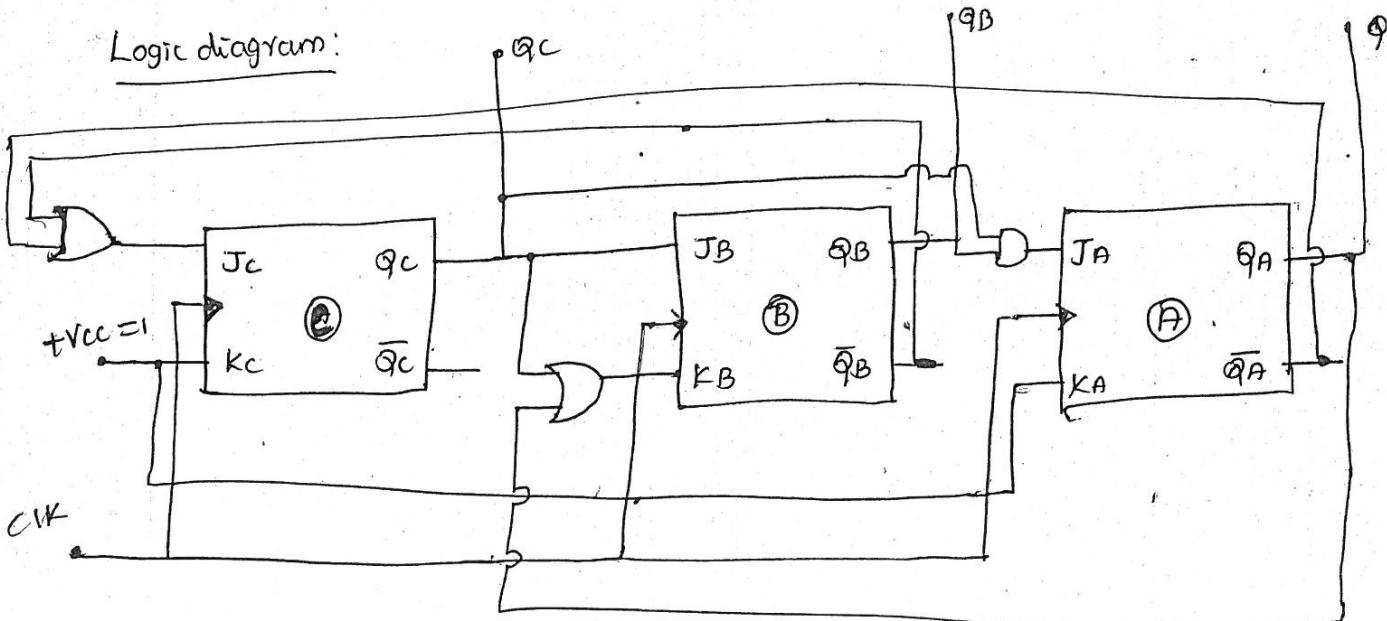
		JC			
		Q _B Q _C	Q _B QC	QBQC	Q _B QC
		0	1	3	2
Q _A	Q _B	1	X	X	1
Q _A	Q _B	1	4	X	0

$$JC = \bar{Q}_A + \bar{Q}_B$$

		KC			
		Q _B Q _C	Q _B QC	QBQC	Q _B QC
		0	1	3 <th>X</th>	X
Q _A	Q _B	X	1	1	X
Q _A	Q _B	1	4	X	X

$$KC = 1$$

Logic diagram:



1) Design of ^{3-bit} up/down synchronous counter.

state Table

CLK	UP	Q _C	Q _B	Q _A	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	

K-map Simplification :-

For TA :-

	Q _B Q _C			
M _{Q_A}	1	0	0	0
M _{Q_A}	1	0	0	0
M _{Q_A}	0	0	1	0
M _{Q_A}	0	0	1	0

$$TA = \overline{M} \overline{Q_B} \overline{Q_C} + M Q_B Q_C$$

For TB :-

	Q _B Q _C			
M _{Q_A}	1	0	0	1
M _{Q_A}	1	0	0	1
M _{Q_A}	0	1	1	0
M _{Q_A}	0	1	1	0

$$TB = \overline{M} \overline{Q_C} + M Q_C$$

For Tc :-

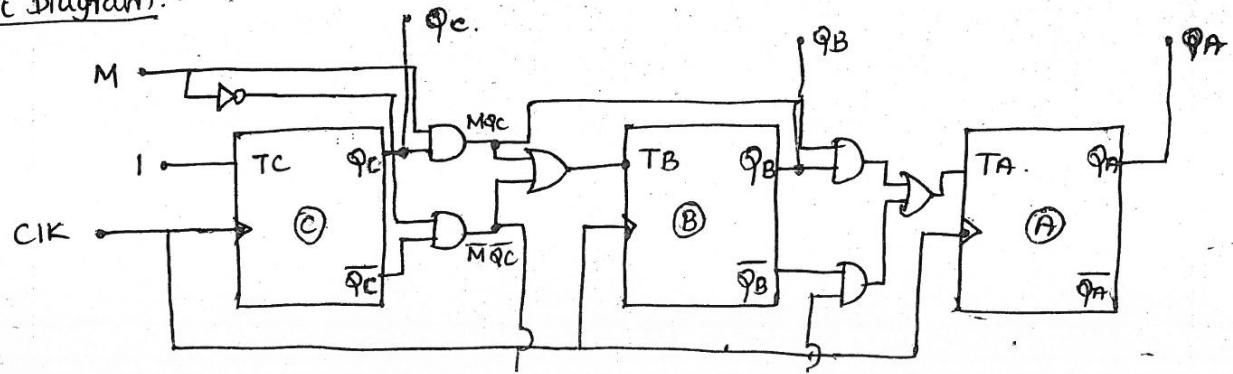
	Q _B Q _C			
M _{Q_A}	1	1	1	1
M _{Q_A}	1	1	1	1
M _{Q_A}	1	1	1	1
M _{Q_A}	1	1	1	1

$$TC = 1$$

Excitation Table

(Input) M	Present state			Next state			Flip Flop - inputs.		
	Q _A	Q _B	Q _C	Q _{A+1}	Q _{B+1}	Q _{C+1}	T _A	T _B	T _C
0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	0	1	1
0	0	1	1	0	1	0	0	0	1
0	1	0	0	0	1	1	1	1	1
0	1	0	1	1	0	0	0	0	1
0	1	1	0	1	0	1	0	1	1
0	1	1	1	1	1	0	0	0	1
1	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
1	0	1	0	0	1	1	0	0	1
1	0	1	1	1	0	0	1	1	1
1	1	0	0	1	0	1	0	0	1
1	1	0	1	1	1	0	0	1	1
1	1	1	0	1	1	1	0	0	1
1	1	1	1	0	0	0	1	1	1

Logic diagram:



2) Design of Synchronous MOD-10 (or) BCD counter using T flip-flop.

Excitation Table

K-map Simplification

For TA			
$\bar{Q}_A \bar{Q}_B$	0, 0	0, 1	0, 2
$\bar{Q}_A Q_B$	0, 4	0, 5	1, 0
$Q_A \bar{Q}_B$	X, 12	X, 13	X, 14
$Q_A Q_B$	0, 8	1, 9	X, 11

$$TA = Q_A \bar{Q}_C \bar{Q}_D + Q_B \bar{Q}_C \bar{Q}_D$$

For TB			
$\bar{Q}_A \bar{Q}_B$	0, 0	0, 1	1, 2
$\bar{Q}_A Q_B$	0, 4	0, 5	1, 6
$Q_A \bar{Q}_B$	X, 12	X, 13	X, 14
$Q_A Q_B$	0, 8	0, 9	X, 11

$$TB = \bar{Q}_C \bar{Q}_D$$

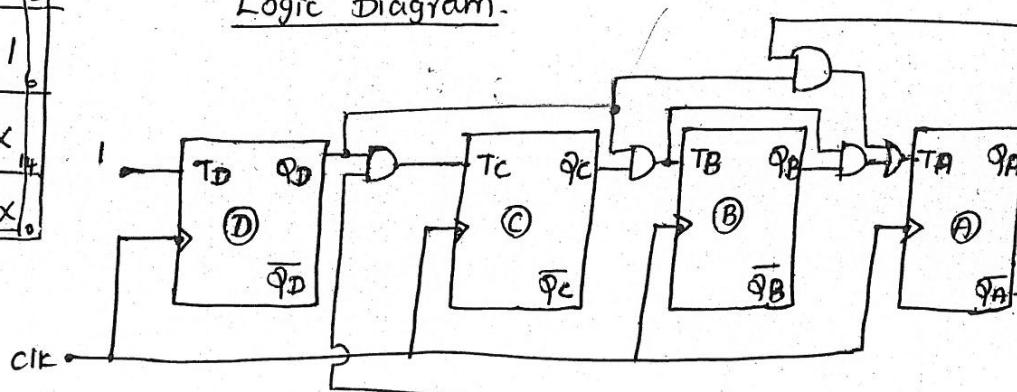
For TC			
$\bar{Q}_A \bar{Q}_B$	0, 0	1, 1	1, 2
$\bar{Q}_A Q_B$	0, 4	1, 5	1, 6
$Q_A \bar{Q}_B$	X, 12	X, 13	X, 14
$Q_A Q_B$	0, 8	0, 9	X, 11

$$TC = \bar{Q}_A \bar{Q}_D$$

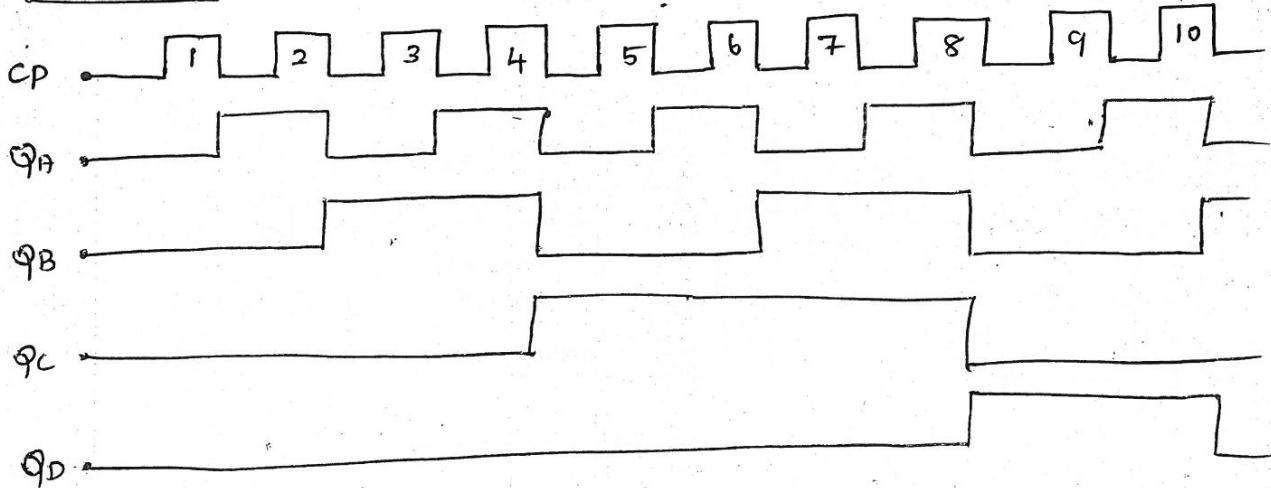
For TD			
$\bar{Q}_A \bar{Q}_B$	1, 0	1, 1	1, 2
$\bar{Q}_A Q_B$	1, 4	1, 5	1, 6
$Q_A \bar{Q}_B$	X, 12	X, 13	X, 14
$Q_A Q_B$	1, 8	1, 9	X, 11

$$TD = 1$$

Logic Diagram.



Waveform:



i) Design a synchronous Mod-6 counter using D flip-flop.

Transition Table

CLK	Present state			Next state			DA	DB	DC
	Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}			
1	0	0	0	0	0	1	0	0	1
2	0	0	1	0	1	0	0	1	0
3	0	1	0	0	1	1	0	1	1
4	0	1	1	1	0	0	1	0	0
5	1	0	0	1	0	1	1	0	1
6	1	0	1	0	0	0	0	0	0
7	1	1	0	x	x	x	x	x	x
8	1	1	1	x	x	x	x	x	x

K-map Simplification.

<u>DA</u>									
$\bar{Q}_B \bar{Q}_C$	$\bar{Q}_B Q_C$	$Q_B \bar{Q}_C$	$Q_B Q_C$						
\bar{Q}_A	0 ₀	0 ₁	1 ₃	0 ₂					
Q_A	1 ₄	0 ₅	x ₇	x ₆					

<u>DB</u>									
$\bar{Q}_B \bar{Q}_C$	$\bar{Q}_B Q_C$	$Q_B \bar{Q}_C$	$Q_B Q_C$						
\bar{Q}_A	0 ₀	1 ₁	0 ₃	1 ₂					
Q_A	0 ₄	0 ₅	x ₇	x ₆					

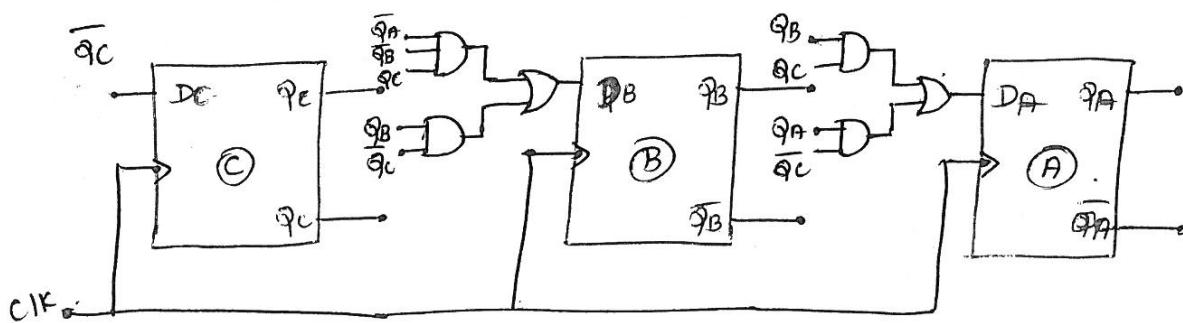
<u>DC</u>									
$\bar{Q}_B \bar{Q}_C$	$\bar{Q}_B Q_C$	$Q_B \bar{Q}_C$	$Q_B Q_C$						
\bar{Q}_A	1 ₀	0 ₁	0 ₃	1 ₂					
Q_A	1 ₄	0 ₅	x ₇	x ₆					

$DA = \bar{Q}_B \bar{Q}_C + \bar{Q}_A \bar{Q}_C$

$DB = \bar{Q}_A \bar{Q}_B \bar{Q}_C + \bar{Q}_B \bar{Q}_C$

$DC = \bar{Q}_C$

Logic Diagram:-

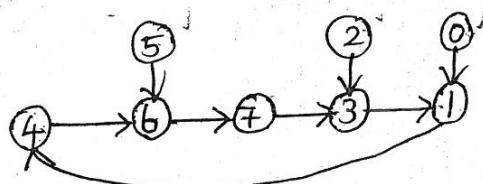


I) Design a synchronous counter for,

$$4 \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow 1 \rightarrow 4 \dots$$

Avoid lockout condition, use JK type design.

Soln:



→ unused state 5, 2, 4, 0 are forced to go into 6, 3, and 1 state, to avoid lockout condition.

Present States			Next states			Flip-flop inputs.					
Q_A	Q_B	Q_C	Q_{A+1}	Q_{B+1}	Q_{C+1}	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	1	0	0	1	X	0	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	1	0	X	X	1	X	0
1	0	0	1	1	0	X	0	1	X	0	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	1	1	X	1	X	0	X	0

K-map Simplification

<u>J_A</u>				
\bar{Q}_A	$\bar{Q}_B\bar{Q}_C$	\bar{Q}_BQ_C	$Q_B\bar{Q}_C$	
\bar{Q}_A	0 ₀	1 ₁	0 ₃	0 ₂
Q_A	X ₄	X ₅	X ₇	X ₆

$J_A = \bar{Q}_A Q_C$

<u>K_A</u>				
\bar{Q}_A	$\bar{Q}_B\bar{Q}_C$	\bar{Q}_BQ_C	$Q_B\bar{Q}_C$	
\bar{Q}_A	X ₀	X ₁	X ₃	X ₂
Q_A	0 ₄	0 ₅	1 ₇	0 ₆

$K_A = \bar{Q}_A Q_C$

<u>J_B</u>				
\bar{Q}_A	$\bar{Q}_B\bar{Q}_C$	\bar{Q}_BQ_C	$Q_B\bar{Q}_C$	
\bar{Q}_A	0 ₀	0 ₁	X ₃	X ₂
Q_A	1 ₄	1 ₅	X ₇	X ₆

$J_B = Q_A$

KB

KB			
\bar{P}_A	X ₀	X ₁	D ₃
P _A	X ₄	X ₅	D ₂

$$KB = \bar{Q}_A Q_C$$

JC

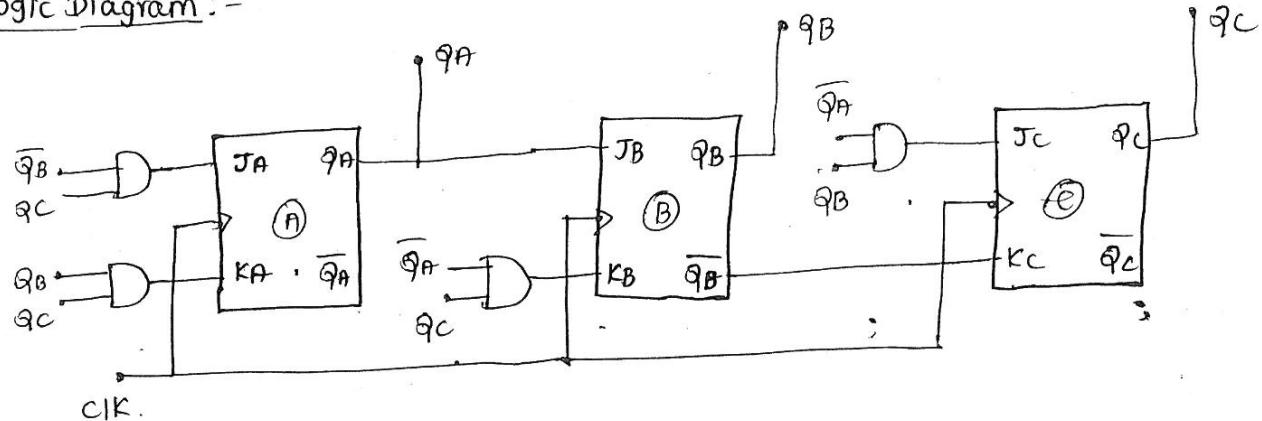
JC			
\bar{Q}_A	1 ₀	X ₁	X ₂
Q _A	0 ₄	X ₅	X ₆

$$JC = \bar{Q}_A + Q_B$$

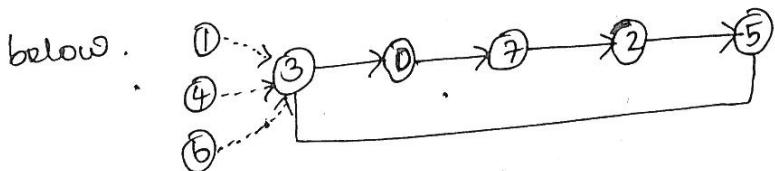
Kc

Kc			
\bar{Q}_A	1 ₀	0 ₃	X ₂
Q _A	X ₄	1 ₅	X ₆

$$Kc = \bar{B}$$

Logic Diagram :-

2) Design synchronous counter using D-flip-flop for the sequence shown below.



Soln:

Q _A	Q _B	Q _C	Q _{A+1}	Q _{B+1}	Q _{C+1}	D _A	D _B	D _C
0	0	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1
0	1	0	1	0	1	1	0	1
0	1	1	0	0	0	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	0	1	1	0	1	1
1	1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1	0

K-map

		$\bar{Q}_A \bar{Q}_B$		$\bar{Q}_A Q_B$		$Q_A \bar{Q}_B$		$Q_A Q_B$	
		0	1	0	1	0	1	0	1
\bar{Q}_C		1	0	0	1	1	0	1	0
Q_A	0	1	0	1	0	1	0	1	0
Q_B	0	0	1	1	1	0	0	1	1
Q_C	1	1	0	0	1	1	0	0	1

$$D_A = \bar{Q}_A \bar{Q}_C$$

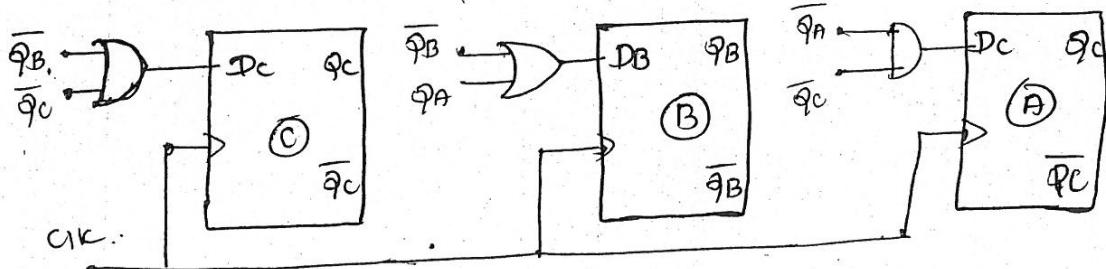
		$\bar{Q}_B \bar{Q}_C$		$\bar{Q}_B Q_C$		$Q_B \bar{Q}_C$		$Q_B Q_C$	
		0	1	0	1	0	1	0	1
\bar{Q}_A		1	0	1	1	0	1	1	0
Q_A	1	1	1	0	1	1	0	1	0
Q_B	1	0	1	1	0	1	1	0	1
Q_C	0	1	0	0	1	0	1	0	1

$$D_B = \bar{Q}_B + Q_A$$

		$\bar{Q}_A \bar{Q}_B$		$\bar{Q}_A Q_B$		$Q_A \bar{Q}_B$		$Q_A Q_B$	
		0	1	0	1	0	1	0	1
\bar{Q}_C		1	0	1	1	0	1	1	0
Q_A	1	1	1	0	1	1	0	1	0
Q_B	1	0	1	1	0	1	1	0	1
Q_C	0	1	0	0	1	0	1	0	1

$$D_C = \bar{Q}_B + \bar{Q}_C$$

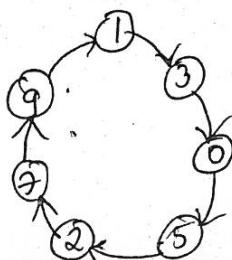
Logic Diagram:-



- (1). Using J-K flip-flop, design a Synchronous counter that has the following Sequence $0 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 0 \dots$

Undesired States 1, 3, 4, 7 must always go to 0 on the next clock pulse.

- (2). Design a Sequence synchronous counter for the following state diag



- (3). Design a Syn. counter which counts all possible odd numbers only

) Design a Syn. counter for the sequence of ... 0 → 2 → 5 → 6 → 0 ...

undesired states 1, 3, 4, 7 must always go to 0 on the next CLK pulse.

Excitation Table

$Q_A Q_B Q_C$	$Q_{A11} Q_{B11} Q_{C11}$	$J_A K_A$	$J_B K_B$	J_C	K_C
0 0 0	0 1 0	0 X	1 X	0 X	
0 0 1	0 0 0	0 X	0 X	X 1	
0 1 0	1 0 1	1 X	X 1	1 X	
0 1 1	0 0 0	0 X	X 1	X 1	
1 0 0	0 0 0	X 1	0 X	0 X	
1 0 1	1 1 0	X 0	1 X	X 1	
1 1 0	0 0 0	X 1	X 1	0 X	
1 1 1	0 0 0	X 1	X 1	X 1	

For JA

$Q_B \bar{Q}_C$	$\bar{Q}_B Q_C$	$Q_B \bar{Q}_C$	$Q_B \bar{Q}_C$
0 0	0 1	0 3	1
X 4	X 5	X 7	X 6

$$JA = Q_B \bar{Q}_C$$

For KA

$\bar{Q}_B \bar{Q}_C$	$\bar{Q}_B \bar{Q}_C$	$Q_B \bar{Q}_C$	$Q_B \bar{Q}_C$
X 0	X 1	X 3	1
1 4	0 5	1 7	1 6

$$KA = \bar{Q}_C + Q_B$$

For KB

$\bar{Q}_B \bar{Q}_C$	$\bar{Q}_B \bar{Q}_C$	$Q_B \bar{Q}_C$	$Q_B \bar{Q}_C$
1 0	0 1	1 3	1
X 4	1 5	X 7	X 6

$$KB = \bar{Q}_A Q_C + Q_A Q_C$$

$$KB = 1$$

For JC

$\bar{Q}_B \bar{Q}_C$	$\bar{Q}_B \bar{Q}_C$	$Q_B \bar{Q}_C$	$Q_B \bar{Q}_C$
0 0	X 1	(X 3)	1
0 4	X 5	1 7	0 6

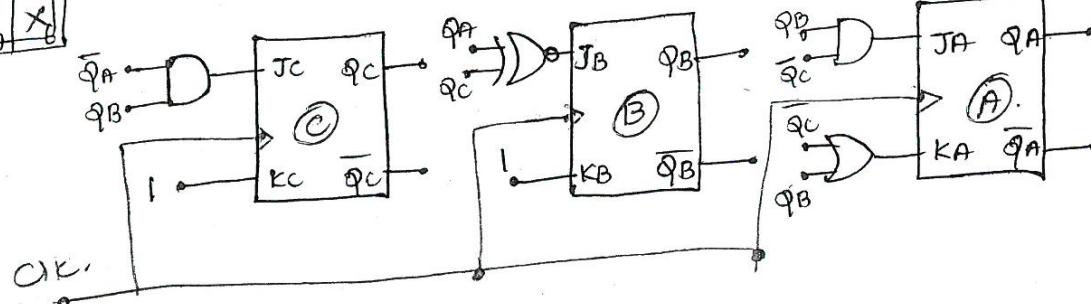
$$JC = \bar{Q}_A \bar{Q}_B$$

For KC

$\bar{Q}_B \bar{Q}_C$	$\bar{Q}_B \bar{Q}_C$	$Q_B \bar{Q}_C$	$Q_B \bar{Q}_C$
X 0	1 1	1 3	X 4
X 4	1 5	1 7	X 6

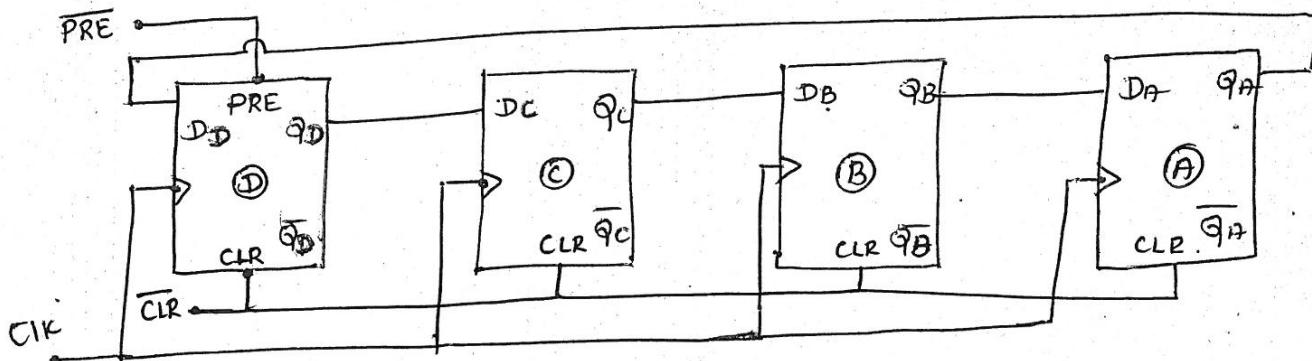
$$KC = 1$$

Logic diagram



Ring Counters

1) Design a 4 bit ring counter.



→ Q output of each stage is connected to the D input of the next stage

and output of last stage is fed back to the input of first stage.

→ The \overline{CLR} followed by \overline{PRE} makes the output of first stage to '1' and

remaining outputs are zero. i.e., Q_D is one & Q_B, Q_C, Q_A are zero.

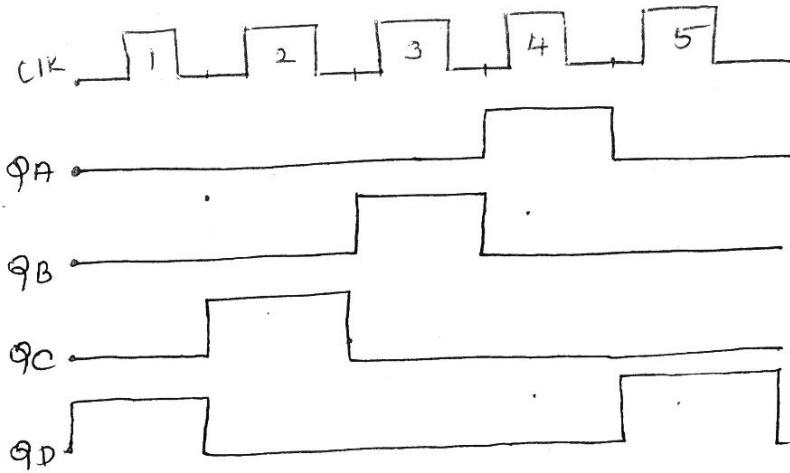
CLK.	Q_A	Q_B	Q_C	Q_D .
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0
5	0	0	0	1

→ The 2nd clock pulse produces $Q_C=1$ and remaining outputs are zero. According to the clock pulses applied at the Clk input CP, a sequence of four states is produced.

→ So, here 1 is always retained in the counter and simply shifted 'around the ring' advancing one state for each clock pulse. In this case four stages of flip-flops are used. so a sequence of four states is produced and repeated.

of flip-flops are used. so a sequence of four states is produced and repeated.

Timing Diagram:-



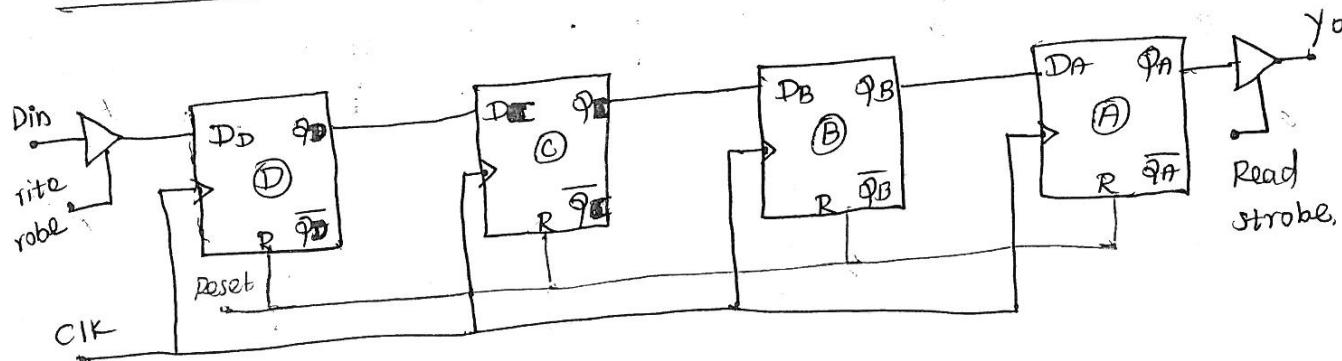
Shift Registers:-

→ The group of flip-flops are connected in cascaded manner, & which is used to store a word → It is called Register.

→ The binary information (data) in a register can be moved from stage to stage within the register (or) into (or) out of the register upon application of the clock pulses. → It is called Shift Registers.

Mode of operation in Shift Registers:-

Serial In Serial Out Shift Register:



→ The write and read operations are in serial form. The input terminal is D_{in} , and its output terminals is y_o .

→ This shift register is a synchronous register, because clock pulse is applied to all flip-flops simultaneously. The output of 1st FF is connected to input of 2nd FF, and the output 2nd FF is connected to input of 3rd FF and so on.

Write operation:- [Not written into registers]

CLK	D_{in}	Write Strobe	Flip-flop outputs			
			QA	QB	QC	QD
Reset	-	0	0	0	0	0
↓	1	1	0	0	0	1
↓	1	1	0	0	1	0
↓	0	1	0	1	1	0
↓	1	1	1	1	0	1

→ Reset all the flip-flops.

→ Make to enable the write strobe signal.

→ Apply data input to D_{in} terminal, one by one.

→ CLK should be applied to clock terminal after applying every data input. → Disable the write strobe signal.

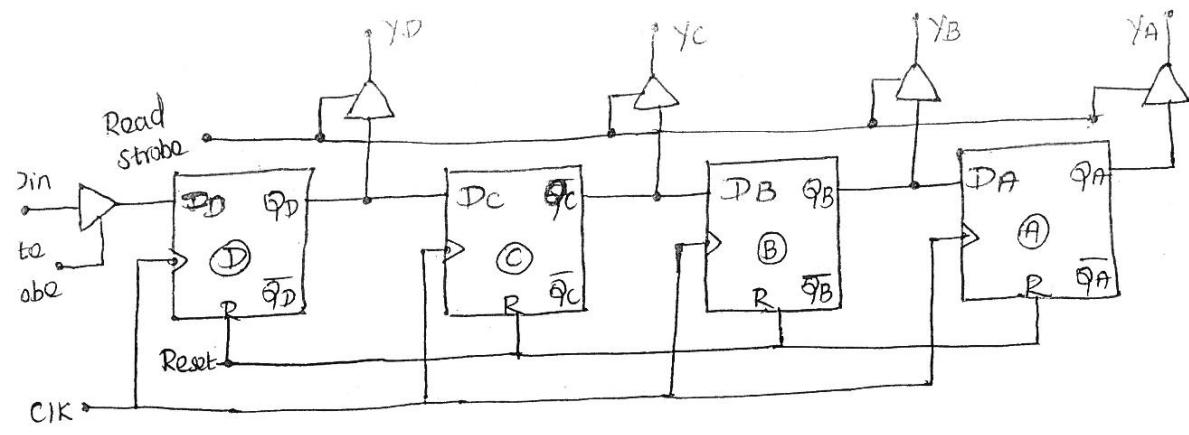
Read operation:- → Enable the read strobe signal.

→ Then apply the clock pulse to the clock input terminal.

→ We can get data one by one in serial manner. and Disable read strobe signal.

Serial-in parallel out :-

→ The D_{in} terminal acts as the input, and y_A, y_B, y_C, y_D terminals as the output.



Write operation:- → Reset all the flip-flops.

→ Make to enable the write strobe signal.

→ Apply the data input one by one & apply the CLK pulses. and Disable the write strobe signal.

CLK input	Din	Write Strobe	Flip-flop outputs			
			QA	QB	QC	QD
Reset	-	0	0	0	0	0
↓	1	1	0	0	0	1
↓	1	1	0	0	1	1
↓	0	1	0	1	1	0
↓	1	1	1	1	0	1

Read operation: → After write operation, we can enable the read strobe signal. and to read data by using the output terminals.

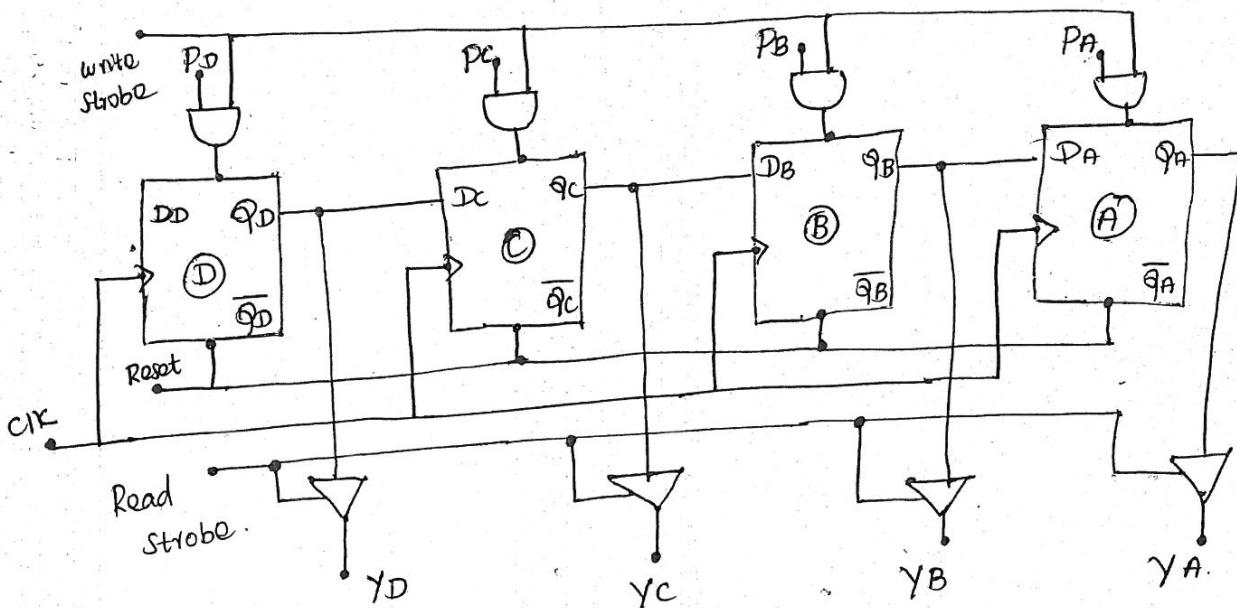
$$Y_A Y_B Y_C Y_D = 1101$$

→ The Disable the read strobe signal.

Parallel in - parallel out:- → This operations both read and write are in parallel form. The clocks are not used for write and read operations.

→ The input terminals are PA PB PC PD and the output terminals are

$$Y_A Y_B Y_C Y_D$$



Write operation :-

- Reset all the flip-flops ($Q_A Q_B Q_C Q_D = 0000$)
- Apply the input data directly to the parallel input terminals,
 $P_A P_B P_C P_D$. [$P_A P_B P_C P_D = 1101$].
- After write operation, the data applied to its i/p terminals are stored in the concerned flip-flops.

Read operation :-

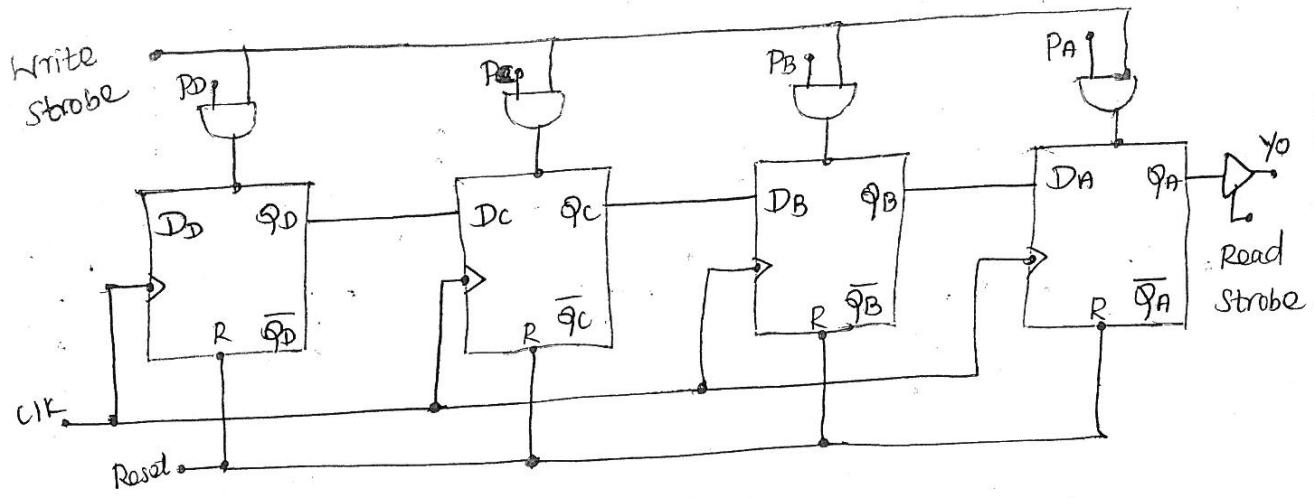
- After the write operation, we can able to read the data directly, after enabling the read strobe signal, the data input already written in to the register is occurred at the output terminals $Y_A Y_B Y_C Y_D$.

$$[Y_A Y_B Y_C Y_D = 1101]$$

Parallel in Serial out :-

- In this type the write operation is in parallel form and the read operation in serial form.

→ The input terminals are P_A P_B P_C P_D and the output terminal is y_0 .



Write operation :- → Reset all the flip-flops.

→ Apply the parallel input to the input terminals P_A P_B P_C P_D .

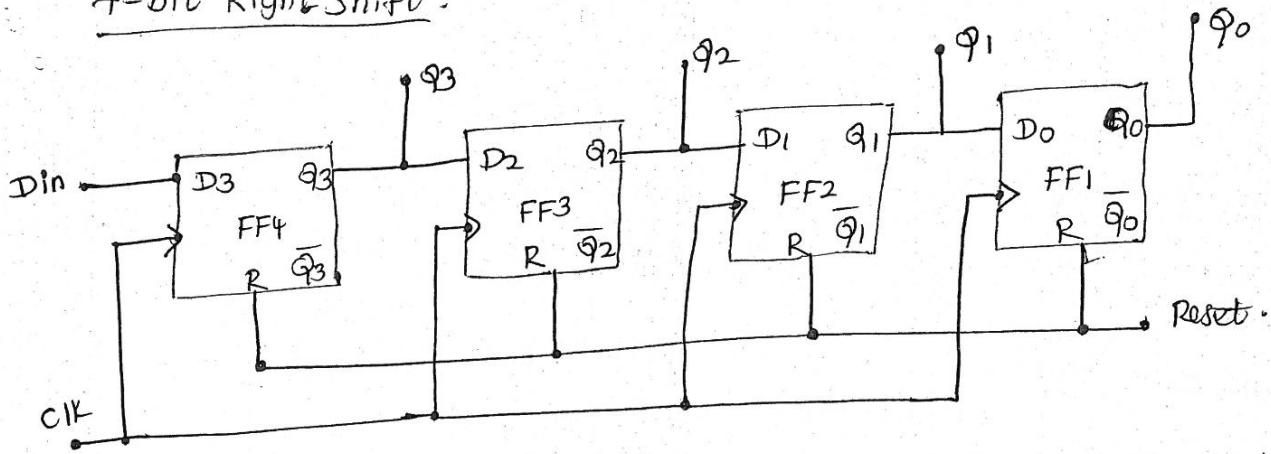
after these operations, the applied input data is stored in the registers.

Read operation :- → After write operation, to read the data by using y_0 output terminal, the clock pulses are needed for read operation.

CLK input	Read Strobe	Q_A	Q_B	Q_C	Q_D	Serial output(y_0)
-	1	1	1	0	1	1 (first data)
↓	1	0	1	1	0	0 (2nd data)
↓	1	0	0	1	1	1 (3rd data)
↓	1	0	0	0	1	1 (4th data)

→ The data are occurred at the output terminal after applying every sequential clock pulses.

4-bit Right Shift :-



→ The actual input is applied to the FF-4 flip flop directly. The output of FF-3 is connected to input of FF-2 and so on.

→ Initially reset all the flip-flops and make the input D_{in} as 1 and apply clk pulses sequentially. During reset condition $Q_3 Q_2 Q_1 Q_0 = 0000$

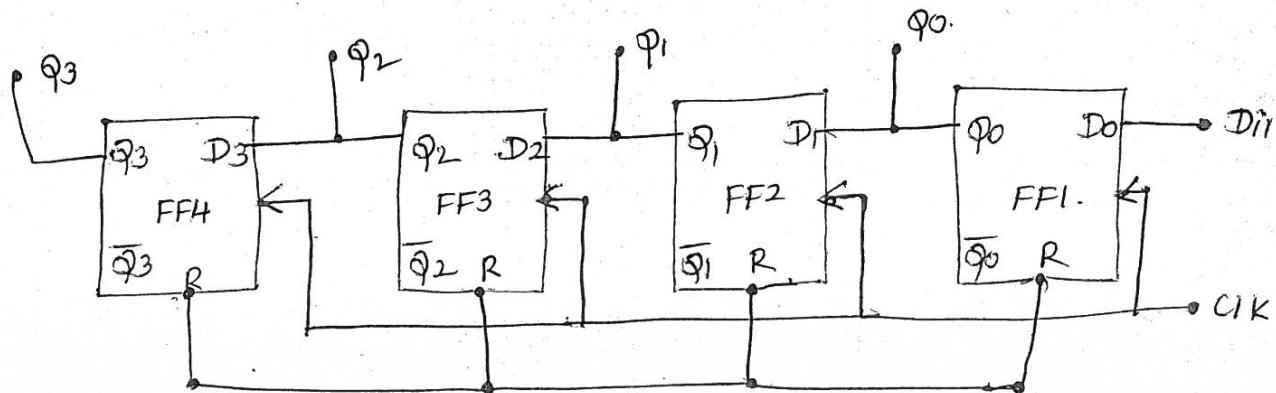
→ After first clock pulse $\Rightarrow Q_3 Q_2 Q_1 Q_0 = 1000$.

→ After 2nd clk pulse $\Rightarrow Q_3 Q_2 Q_1 Q_0 = 1100$

→ After 3rd clk pulse $\Rightarrow Q_3 Q_2 Q_1 Q_0 = 1110$ and finally after

4th clk pulse applied $\Rightarrow Q_3 Q_2 Q_1 Q_0 = 1111$.

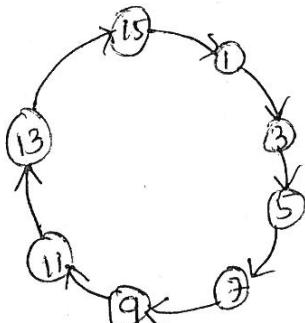
4-bit Left-Shift :-



- The data input is applied to FF-1 only. The output of FF-1 is connected to the input of FF-2, and so on.
- Initially all the flip-flops are going to reset, and the output of $Q_3 Q_2 Q_1 Q_0 = 0000$. Then D_{in} input is 1, and apply clk pulses simultaneously.
- During first clock pulse the output is $Q_3 Q_2 Q_1 Q_0 = 0001$ and during 2nd clock pulse $Q_3 Q_2 Q_1 Q_0 = 0011$, and after 3rd and 4th clock pulses the outputs are 0111 and 1111 respectively.

- 1). Design a 4-bit syn. counter using D-FF, which counts all possible odd numbers only.

State diagram.



Excitation Table:

Q_A	Q_B	Q_C	Q_D	Q_{A+1}	Q_{B+1}	Q_{C+1}	Q_{D+1}	D_A	D_B	D_C	D_D
0	0	0	0	x	x	x	x	0	0	1	1
0	0	0	1	0	0	1	1	x	x	x	x
0	0	1	0	x	x	x	x	0	1	0	1
0	0	1	1	0	1	0	1	x	x	x	x
0	1	0	0	x	x	x	x	x	x	1	1
0	1	0	1	0	1	1	1	0	1	1	1
0	1	1	0	x	x	x	x	x	x	x	x
0	1	1	1	1	0	0	1	1	0	0	1
1	0	0	0	x	x	x	x	x	x	x	x
1	0	0	1	1	0	1	1	1	0	1	1
1	0	1	0	x	x	x	x	x	x	x	x
1	0	1	1	1	1	0	1	1	1	0	1
1	1	0	0	x	x	x	x	x	x	x	x
1	1	0	1	1	1	1	1	1	1	1	1
1	1	1	0	x	x	x	x	x	x	x	x
1	1	1	1	0	0	0	1	0	0	0	1

K-map Simplification

FOR DA

	$\overline{Q}_C Q_D$	$\overline{Q}_C Q_D$	$Q_C \overline{Q}_D$	$Q_C Q_D$
$\overline{Q}_A \overline{Q}_B$	X ₀	0 ₁	0 ₃	X ₂
$\overline{Q}_A Q_B$	X ₄	0 ₅	1 ₇	X ₆
$Q_A \overline{Q}_B$	X ₁₂	1 ₁₃	0 ₁₅	X ₁₄
$Q_A Q_B$	X ₈	1 ₉	1 ₁₁	X ₁₀

$$D_A = Q_A \overline{Q}_C + Q_A \overline{Q}_B + \overline{Q}_A Q_B Q_C$$

FOR DC

	$\overline{Q}_C Q_D$	$\overline{Q}_C Q_D$	$Q_C \overline{Q}_D$	$Q_C Q_D$
$\overline{Q}_A \overline{Q}_B$	X ₀	1 ₁	0 ₃	X ₂
$\overline{Q}_A Q_B$	X ₄	1 ₅	0 ₇	X ₆
$Q_A \overline{Q}_B$	X ₁₂	1 ₁₃	0 ₁₅	X ₁₄
$Q_A Q_B$	X ₈	1 ₉	0 ₁₁	X ₁₀

$$D_C = \overline{Q}_C$$

FOR DB

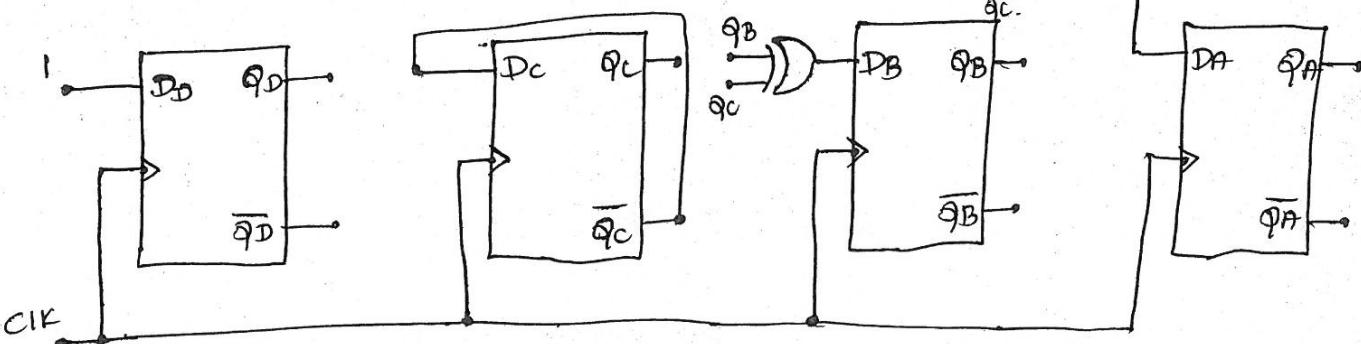
	$\overline{Q}_C Q_D$	$\overline{Q}_C Q_D$	$Q_C \overline{Q}_D$	$Q_C Q_D$
$\overline{Q}_A \overline{Q}_B$	X ₀	0 ₁	1 ₃	X ₂
$\overline{Q}_A Q_B$	X ₄	1 ₅	0 ₇	X ₆
$Q_A \overline{Q}_B$	X ₁₂	1 ₁₃	0 ₁₅	X ₁₄
$Q_A Q_B$	X ₈	0 ₉	1 ₁₁	X ₁₀

$$D_B = Q_B \overline{Q}_C + \overline{Q}_B Q_C$$

FOR DD

	$\overline{Q}_C Q_D$	$\overline{Q}_C Q_D$	$Q_C \overline{Q}_D$	$Q_C Q_D$
$\overline{Q}_A \overline{Q}_B$	X ₀	1 ₁	1 ₃	X ₂
$\overline{Q}_A Q_B$	X ₄	1 ₅	1 ₇	X ₆
$Q_A \overline{Q}_B$	X ₁₂	1 ₁₃	1 ₁₅	X ₁₄
$Q_A Q_B$	X ₈	1 ₉	1 ₁₁	X ₁₀

$$D_D = 1$$



- 1) MOD-6 Synchronous Counter Using T - Flip-flop.
- 2) Design a Syn. counter that can count only one digit prime number using JK flip-flop.
- 3) Design a counter which counts the following Sequence,
018, 12, 10, 14, 19, 13, 11, 15, 08, 12.....
- 4) Design a Syn. counter it counts primary numbers between 0 to 15 using JK flip-flop.

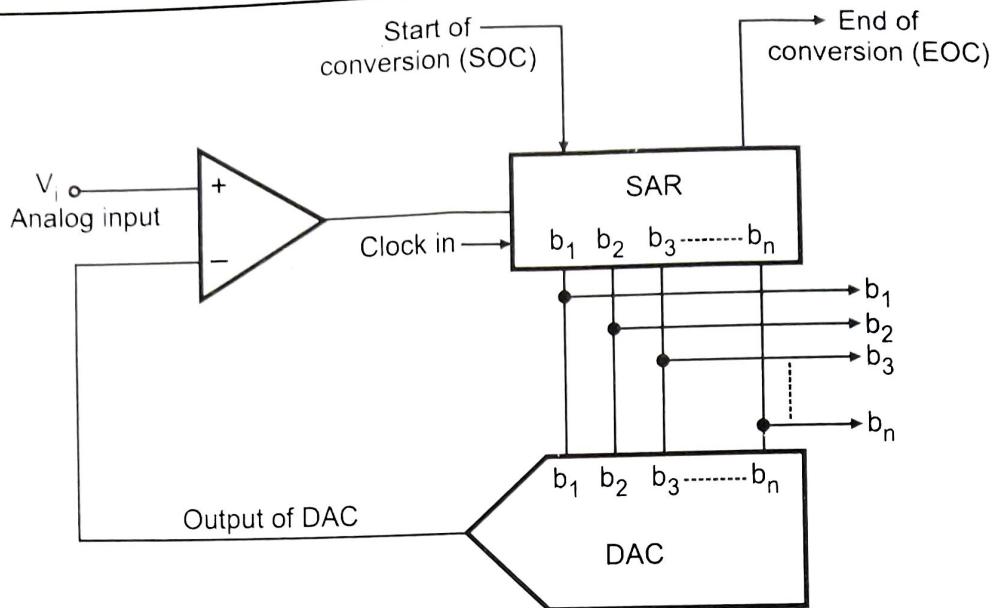


Fig. 7.11 Block diagram of successive approximation A/D converter

Operation :

The searching code process in successive approximation method is similar to weighing an unknown material with a balance scale and a set of standard weights. Let us assume that we have 1 kg, 2 kg and 4 kg weights (SAR) plus a balance scale (comparator and DAC). Now we will see the successive approximation analogy for 3-bit ADC.

Refer Fig. 7.11 and 7.12. The analog voltage V_{in} is applied at one input of comparator. On receiving start of conversion signal (SOC) successive approximation register sets 3-bit binary code 100_2 ($b_2 = 1$) as an input of DAC. This is similar process of placing the unknown weight on one platform of the balance and 4 kg weight on the other. The DAC converts the digital word 100 and applies it equivalent analog output at the second input of the comparator. The comparator then compares two voltages just like comparing unknown weight with 4 kg weight with the help of balance scale. If the input voltage is greater than the analog output of DAC, successive approximation register keeps $b_2 = 1$ and makes $b_1 = 1$ (addition of 2 kg weight to have total 6 kg weight) otherwise it resets $b_2 = 0$ and makes $b_1 = 1$ (replacing 2 kg weight). The same process is repeated for b_1 and b_0 . The status of b_0, b_1 and b_2 bits gives the digital equivalent of the analog input.

Fig. 7.12 illustrates the process we have just discussed.

The dark lines in the Fig. 7.12 shows setting and resetting actions of bits for input voltage 5.2 V, on the basis of comparison. It can be seen from the Fig. 7.12 that one clock pulse is required for the successive approximation register to compare each bit. However an additional clock pulse is usually required to reset the register prior to performing a conversion.

The time for one analog to digital conversion must depend on both the clock's period T and number of bits n . It is given as,

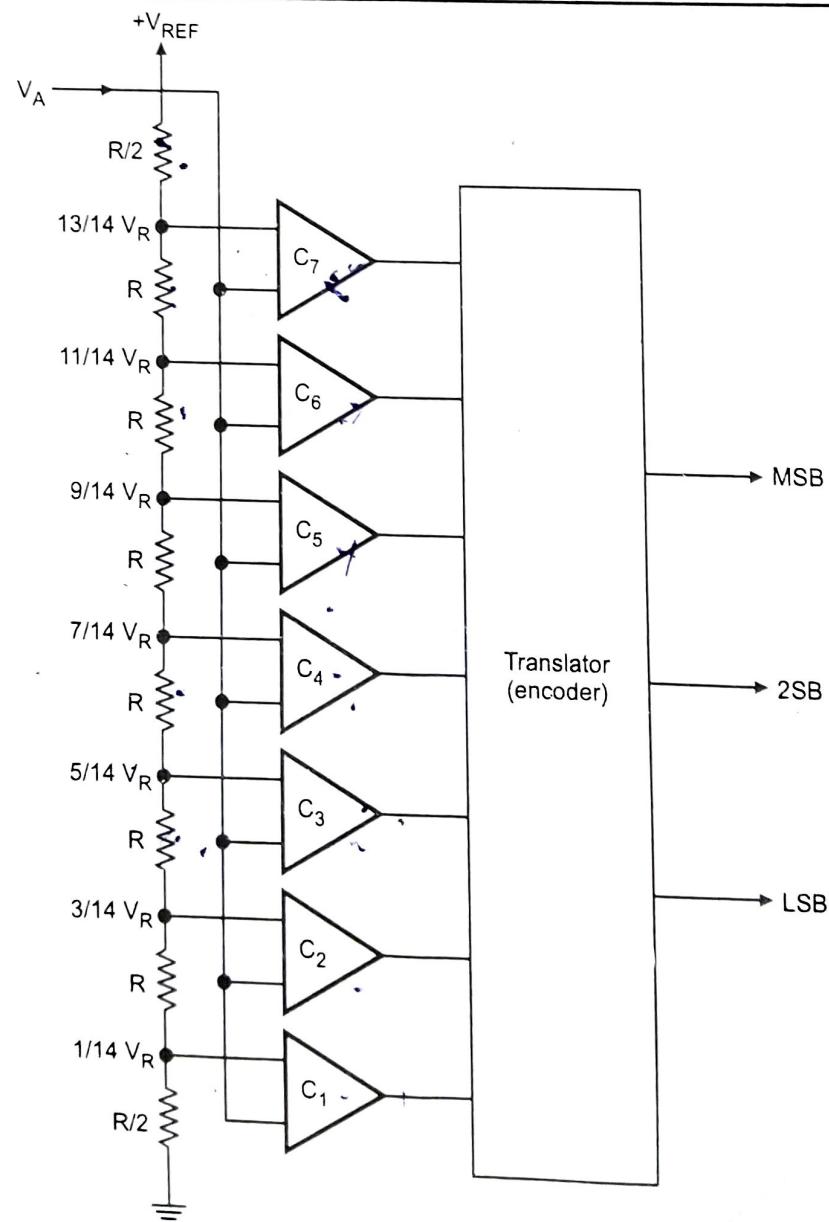
$$T = \frac{1}{f} = \frac{1}{1 \times 10^6} = 1 \text{ sec} \quad n = 8$$

$$T_C = T(n + 1) = 1(8 + 1) = 9 \text{ sec.}$$

7.12 Flash ADC

When system designs call for the highest speed available, flash-type A/D converters (ADCs) are likely to be the right choice. They get their names from their ability to do the conversion very rapidly. Flash A/D converters, also known as a simultaneous or parallel comparator ADC, because the fast conversion speed is accomplished by providing $2^n - 1$ comparators and simultaneously comparing the input signal with unique reference levels spaced 1 LSB apart.

The Fig. 7.13 shows 3-bit flash A/D converter. For this ADC, seven ($2^3 - 1$) comparators are required. As shown in the Fig. 7.13, one input of each comparator is connected to the input signal and other input to the reference voltage level generated by the reference voltage divider. The reference voltage (V_{REF}) is equal to the full scale input signal voltage. The manner in which the flash A/D converter performs quantization is relatively simple.



7.10 Dual Slope ADC

Dual slope conversion is an indirect method for A/D conversion where an analog voltage and a reference voltage are converted into time periods by an integrator, and then measured by a counter. The speed of this conversion is slow but the accuracy is high.

Fig. 7.9 shows a typical dual slope converter circuit. It consists of integrator (ramp generator), comparator, binary counter, output latch and reference voltage. The ramp generator input is switched between the analog input voltage V_i and a negative reference voltage, $-V_{REF}$. The analog switch is controlled by the MSB of the counter. When the MSB is a logic 0, the voltage being measured is connected to the ramp generator input. When MSB is logic 1, the negative reference voltage is connected to the ramp generator.

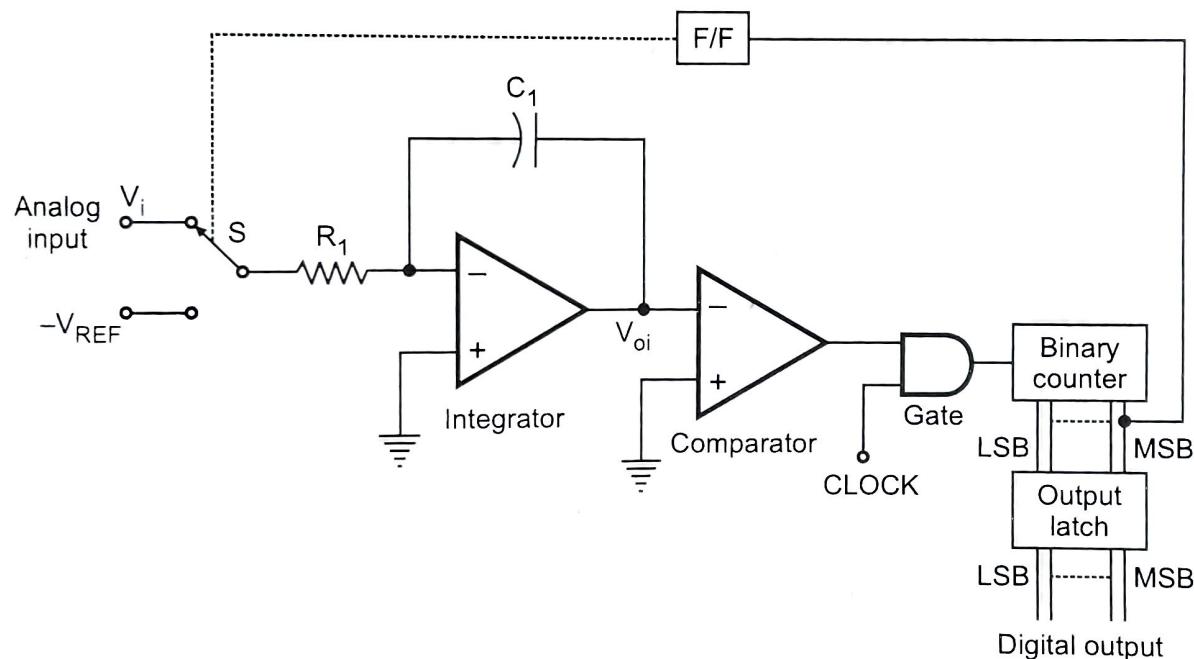
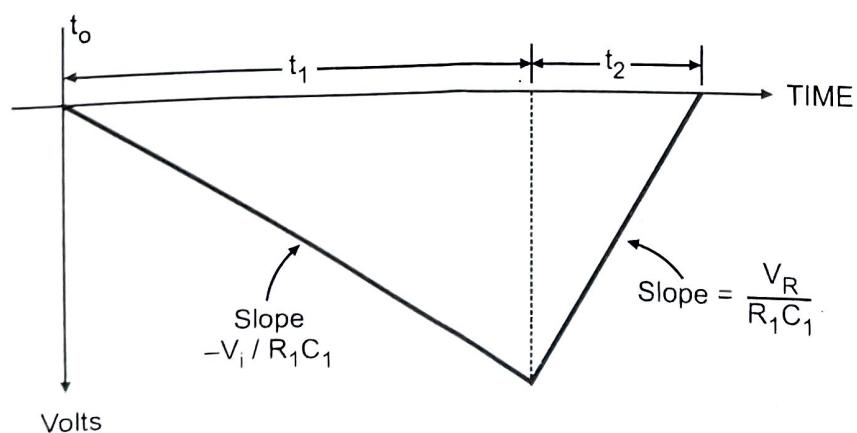


Fig. 7.9 Dual slope A/D converter



6.4 Basic Conversion Techniques

There are mainly two techniques used for analog to digital conversion

- Binary weighted resistor D/A converter
- R/2R ladder D/A converter

6.4.1 Binary Weighted Resistor D/A Converter

The binary weighted resistor DAC uses an op-amp to sum n binary weighted currents derived from a reference voltage V_R via current scaling resistors $2R$, $4R$, $8R$, ..., $2^n R$, as shown in the Fig. 6.3.

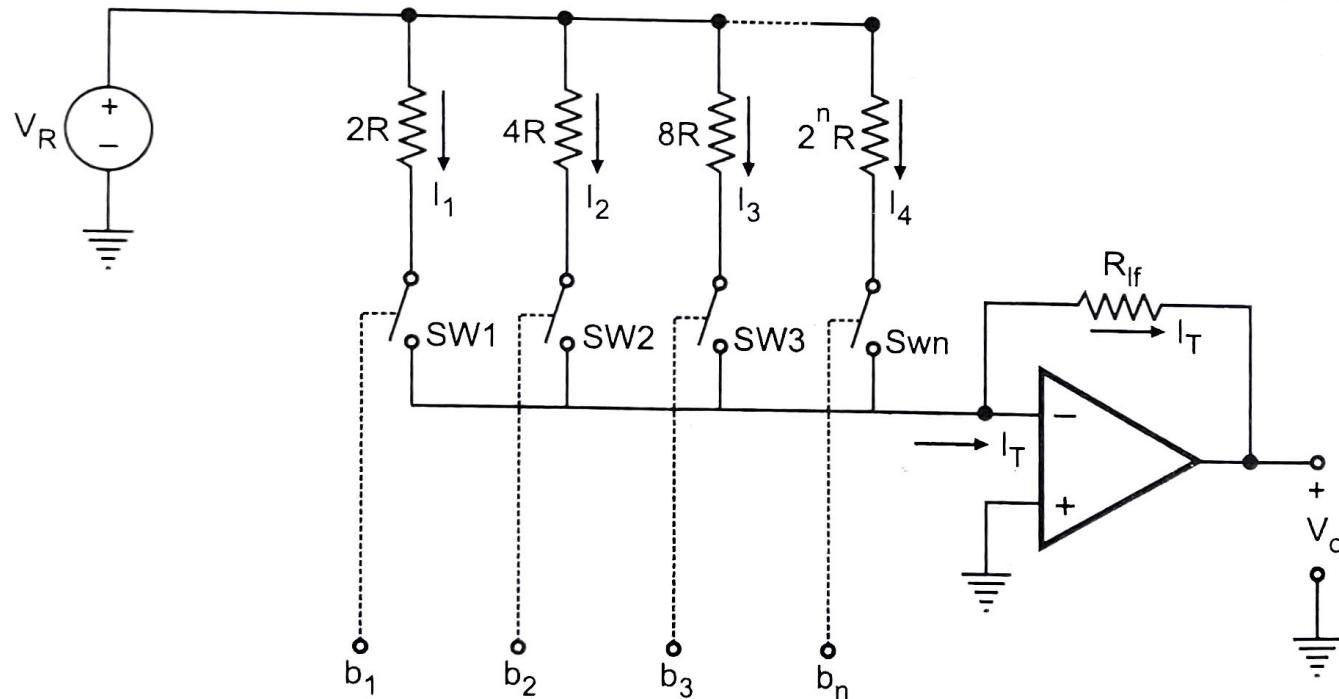


Fig. 6.3 Binary weighted resistor DAC

As shown in the Fig. 6.3, switch positions are controlled by the digital inputs. When digital input is logic 1, it connects the corresponding resistance to the reference voltage V_R ; otherwise it leaves resistor open. Therefore,

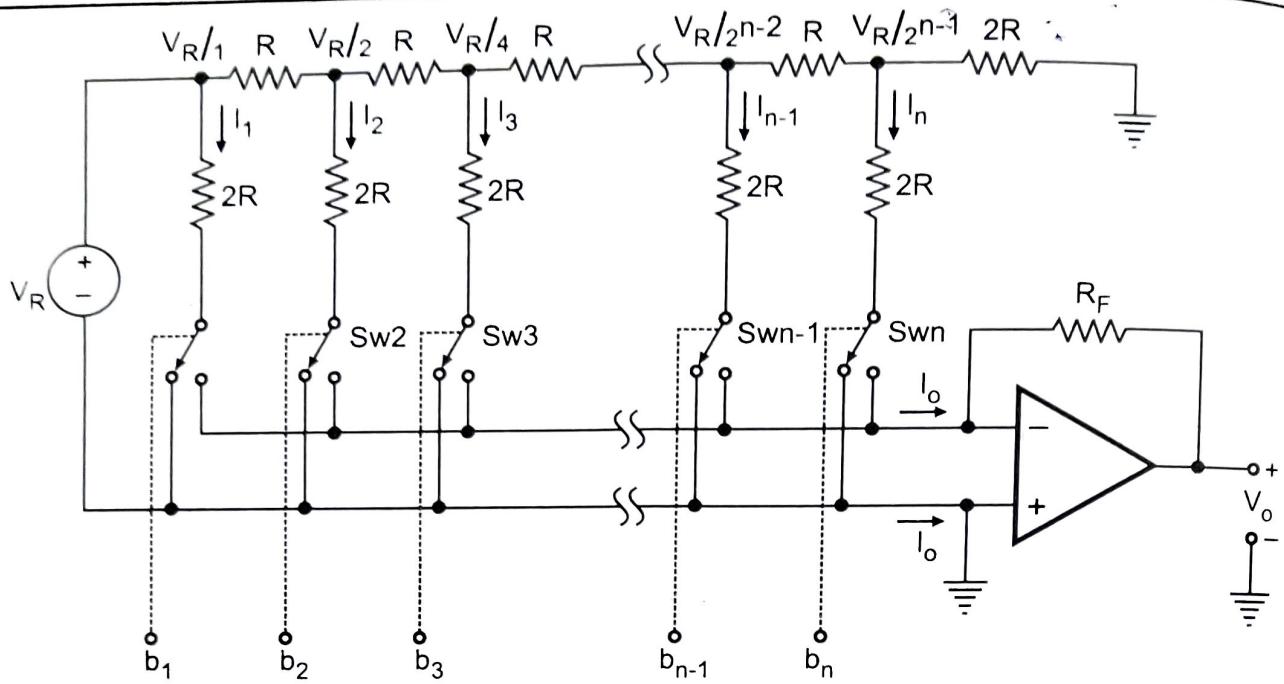


Fig. 6.4 Inverted R/2R ladder DAC

Voltage scaling requires an additional set of voltage dropping series resistances between adjacent nodes, as shown in the Fig. 6.4.

Here, each bit of the binary word connects the corresponding switch either to ground or to the inverting input terminal of the op-amp which is at the virtual ground. Since both the positions of switches are at ground potential, the current flowing through resistances is constant and it is independent of switch position. These currents can be given as

$$I_1 = \frac{V_R}{2R} \quad \dots (6.6)$$

$$I_2 = \frac{V_R/2}{2R}$$

$$= \frac{V_R}{4R} = \frac{I_1}{2}$$

$$I_3 = \frac{V_R/4}{2R}$$

$$= \frac{V_R}{8R}$$

$$= \frac{I_1}{4}$$

$$I_n = \frac{V_R/2^n - 1}{2R}$$

$$= \frac{I_1}{2^n - 1} \quad \dots (6.7)$$

We know that, V_o is given as

$$V_o = -I_T R_f \quad \dots (6.8)$$

6.4.3 R/2R Ladder D/A Converter

In this type, reference voltage is applied to one of the switch positions, and other switch position is connected to ground, as shown in the Fig. 6.6.

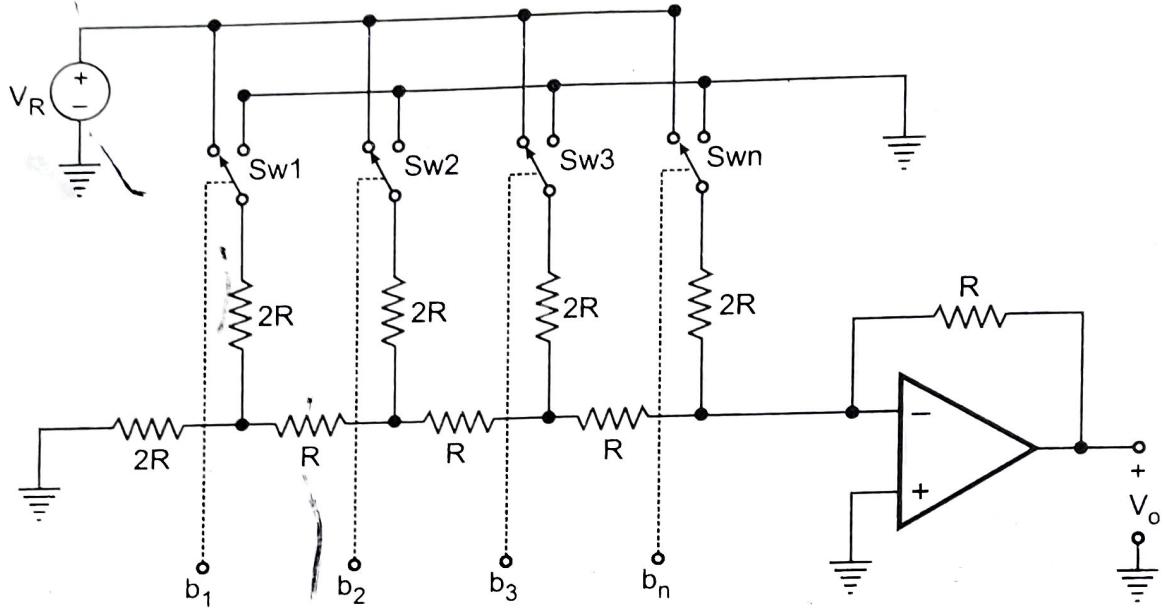


Fig. 6.6 R/2R Ladder D/A converter

Let us consider 3-bit R/2R ladder DAC with binary input 001, as shown in the Fig. 6.7.

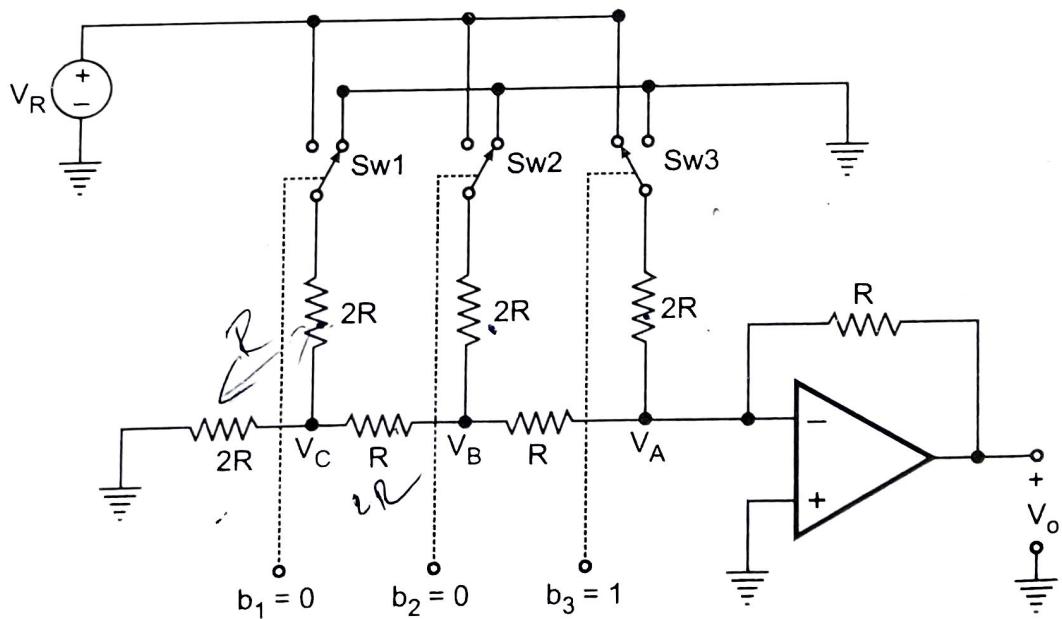


Fig. 6.7 3-bit R/2R ladder DAC

Reducing above network to the left by Thevenin's theorem we get,

Solution

$$\text{Since, } V_a = V_R(N/2^n)$$

So the digital count $N = 2^n$ ($V_a/V_R = 65536$ (4.129 V/8 V) = 33825 for which the binary equivalent is 10000100001000001.

10.4 DAC/ADC SPECIFICATIONS

Both D/A and A/D converters are available with wide range of specifications. The various important specifications of converters generally specified by the manufacturers are analyzed.

Resolution: The resolution of a converter is the smallest change in voltage which may be produced at the output (or input) of the converter. For example, an 8-bit D/A converter has $2^8 - 1 = 255$ equal intervals. Hence the smallest change in output voltage is (1/255) of the full scale output range. In short, the resolution is the value of the LSB.

$$\text{Resolution (in volts)} = \frac{V_{FS}}{2^n - 1} = 1 \text{ LSB increment} \quad (10.8)$$

However, resolution is stated in a number of different ways. An 8-bit DAC is said to have

- : 8 bit resolution
- : a resolution of 0.392 of full-scale
- : a resolution of 1 part in 255

Similarly, the resolution of an A/D converter is defined as the smallest change in analog input for a one bit change at the output. As an example, the input range of an 8-bit A/D converter is divided into 255 intervals. So the resolution for a 10 V input range is 39.22 mV (= 10 V/255). Table 10.1 gives the resolution for 6–16 bit DACs.

Table 10.1 Resolution for 6–16 bit DACs

Bits	Intervals	LSB size (% of Full Scale)	LSB size (10 V Full Scale)
6	63	1.588	158.8 mV
8	256	0.392	39.2 mV
10	1023	0.0978	9.78 mV
12	4095	0.0244	2.44 mV
14	16383	0.0061	0.61 mV
16	65535	0.0015	0.15 mV

Linearity: The linearity of an A/D or D/A converter is an important measure of its accuracy and tells us how close the converter output is to its ideal transfer characteristics. In an ideal DAC, equal increment in the digital input should produce equal increment in the analog output and the transfer curve should be linear. However, in an actual DAC, output voltages do not fall on a straight line because of gain and offset errors as shown by the solid line curve in Fig. 10.17. The static performance of a DAC is determined by

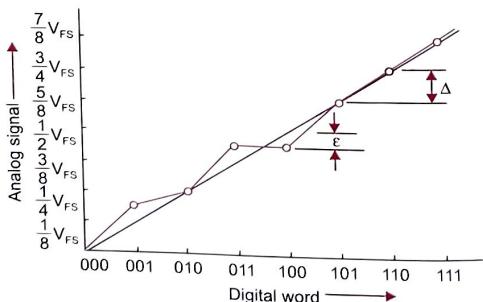


Fig. 10.17 Linearity error for 3-bit DAC

fitting a straight line through the measured output points. The linearity error measures the deviation of the actual output from the fitted line and is given by ϵ/Δ as shown in Fig. 10.17. The error is usually expressed as a fraction of LSB increment or percentage of full-scale voltage. A good converter exhibits a linearity error of less than $\pm (1/2)$ LSB.

Accuracy: Absolute accuracy is the maximum deviation between the actual converter output and the ideal converter output. Relative accuracy is the maximum deviation after gain and offset errors have been removed. Data sheets normally specify relative accuracy increments or percentage of full scale voltage.

Monotonicity: A monotonic DAC is the one whose analog output increases for an increase in digital input. Figure 10.18 represents the transfer curve for a non-monotonic DAC, since the output decreases when input code changes from 001 to 010. A monotonic characteristic is essential in certain applications, otherwise oscillations may result. In successive approximation converters a non-monotonic characteristic may lead to missing codes.

If a DAC has to be monotonic, the error should be less than $\pm(1/2)$ LSB at each output level. All the commercially available DACs are monotonic because the linearity error never exceeds $\pm(1/2)$ LSB at each output level.

Settling time: The most important dynamic parameter is the settling time. It represents the time it takes for the output to settle within a specified band $\pm(1/2)$ LSB of its final value following a code change at the input (usually a full scale change). It depends upon the switching time of the logic circuitry due to internal parasitic capacitances and inductances. Settling time ranges from 100 ns to 10 μ s depending on word length and type of circuit used.

Stability: The performance of converter changes with temperature, age and power supply variations. So all the relevant parameters such as offset, gain, linearity error and monotonicity must be specified over the full temperature and power supply ranges.

A brief overview of ADC and DAC selection guide is given below:

A/D converters:

AD 7520/AD 7530	10-bit binary multiplying type
AD 7521/AD 7531	12-bit binary multiplying type
ADC 0800/0801/0802	8-bit ADC

D/A converters:

DAC 0800/0801/0802	8-bit DAC
DAC 0830/0831/0832	microprocessor compatible 8-bit DAC
DAC 1200/1201	12-bit DAC
DAC 1208/1209/1210	12-bit microprocessor compatible DAC

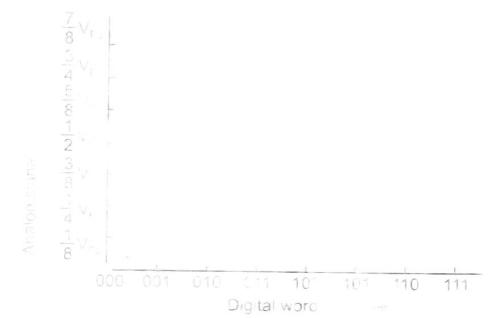


Fig. 10.18 A non-monotonic 3-bit DAC