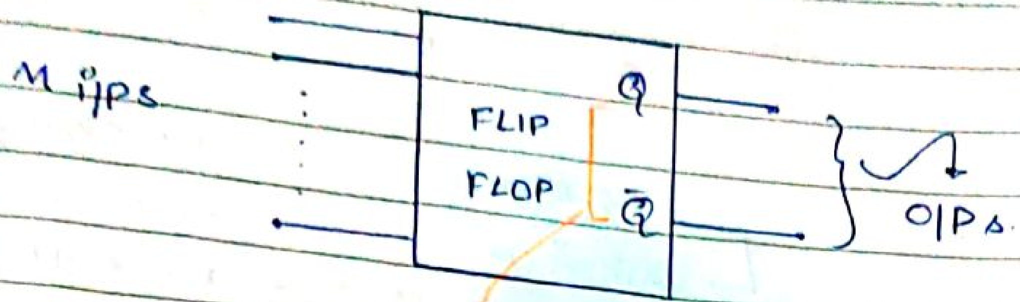
Sequential Logic Circuits

* Flip - Flop : Memory Unit

Block diagram :



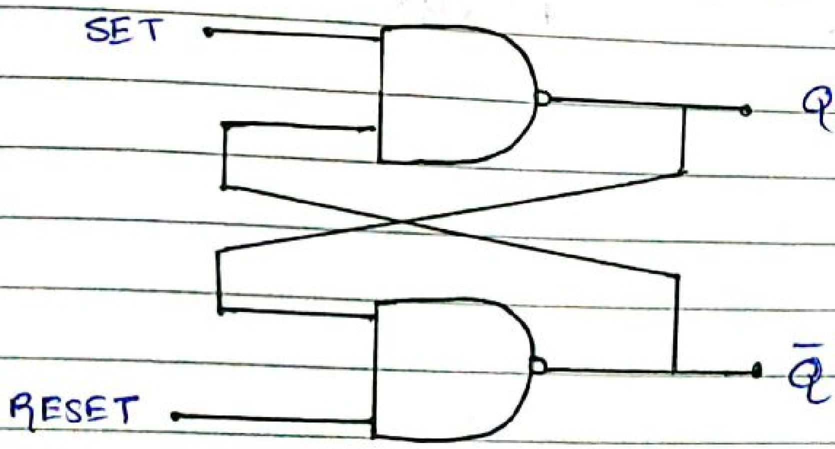
Must be
Compliment of each other.

If
$Q = 1$, ff's O/P is high
SET STATE
$\bar{Q} = 0$

Conclusion on
basis of Q.

If $Q = 0$, LOW or RESET or CLEAR.
$\bar{Q} = 1$

* NAND LATCH ⟶ AS MEMORY



R-S FF
USING NAND
LATCH

0 0̄, 1, 1 ②
1 1 0 0 0 getting
Assume 1, 1, 1 / 0, 0, 0 getting → O/P

Assume 0, 1, 1 ①
↳ Not complimentary + invalid (0, 0)
1, 1, 1 getting

Not changing memory

I/P		O/P			
		PRESENT STATE		NEXT STATE	
SET	RESET	Q_n	$\bar{Q}_n$	Q_{n+1}	$\bar{Q}_{n+1}$
0	0	Invalid		Invalid	
0	1	1	0	1	0
1	0	0	1	0	1
1	1	No change.		No change	

← Acting as Memory Element

Value is Repeating i.e we are getting what initially was stored.

↳ When we assume Q as 1 we are getting 1 everytime. Illy when we assume Q as 0 we are getting 0 everytime

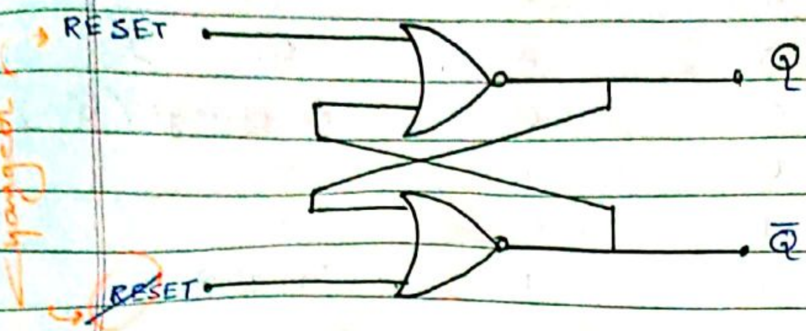
PROFESIONAL : ☺
TABLE

I/P		O/P
S	R	Q
0	0	Invalid
0	1	1
1	0	0
1	1	No change. (Q_n)

$Q = \bar{Q}$ → getting

} MEMORY

* NOR LATCH.

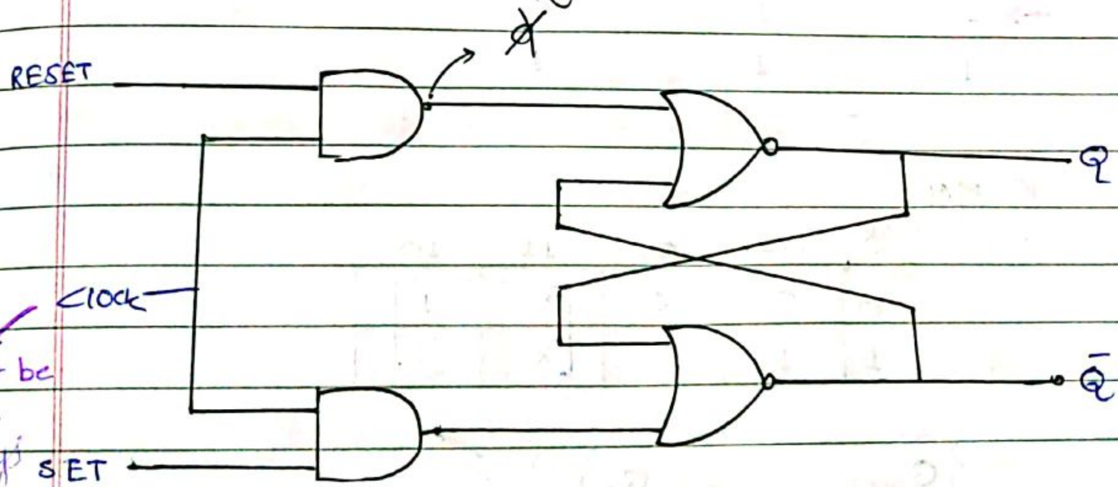


R-S FF USING NOR LATCH

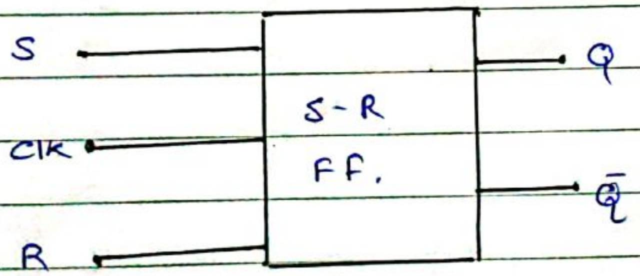
I/Ps		O/Ps.
S	R	Q_m
0	0	NO change. (Q_m)
0	1	0
1	0	1
1	1	Invalid

} MEMORY

* RS Flip - Flop Using ~~NAND~~ and NOR Gates.



Must be high always else NOT WORK.



X ⇒ Don't Care

T.T of Clocked SR

Clk	S	R	Q_m
0	X	X	Q_m
1	0	0	No change (Q_m)
1	0	1	0
1	1	0	1
1	1	1	Invalid.

To isolate Flip Flop

$R = \bar{S}$ v.v.

Characteristic Table

Given

CURRENT STATE

NEXT STATE → Find

Q_m	S	R	Q_{m+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Check only for no change cond^n

K-MAP

Q_m \ SR	00	01	11	10
0	0	0	X	1
1	1	0	X	1

$$\boxed{Q_{m+1} = S + Q_m\bar{R}}$$

Scanned by CamScanner

* Excitation Table for clocked F.F. (R-S) (From C. Tab

ET is a Table in which, we want O/P and we have to decide i/p.

i.e Given : O/P

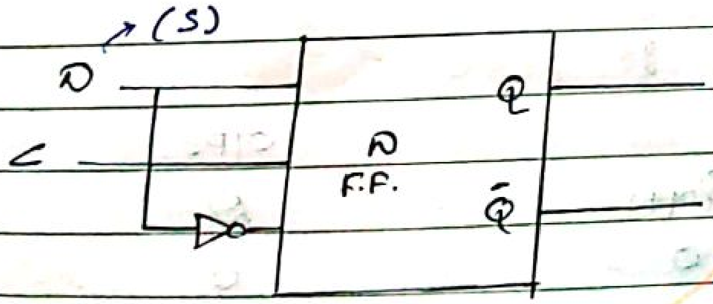
Find : I/P

I/P		O/P	
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	X0

R → DON'T CARE

S, R: From characteristic Table.

* D- Flip - Flop



Using SR FF

Clock	I/Ps	Q_n
0	X	Q_n
1	0	Q = 0
1	1	Q = 1.
~~1~~	~~1~~	~~Invalid~~

D = 0
S = 0 R = 1
D = 1.
S = 1 R = 0

* Characteristic Table

PREVIOUS/CURRENT STATE I/P

Q_n	D	O/P Q_{n+1} → NEXT STATE
0	0	0
0	1	1
1	0	0
1	1	1

When $D=0$ we get zero (0) ~~no change~~ ⇒ whatever Q is ~~it is repeated~~

When $D=1$ we always get 1

From C. Table

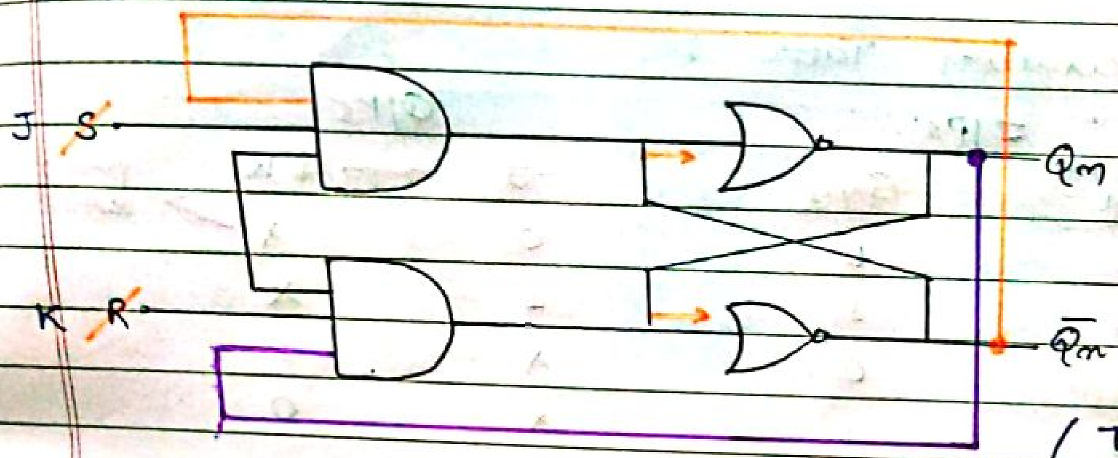
$$\boxed{Q_{n+1} = D}$$

* Excitation Table :- From C.Table

I/Ps		O/Ps
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

$\{Q_{n+1} = D\}$

* JK flip-flop.



(To overcome 1 1 Invalid Cond^n)

T.T for JK F.F.

Clock.	J	K.	Q_{n+1}.
0	X	X	Q_n
1.	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1.	$\overline{Q_n}$

Q_n : 0 1 0 1 ...

Q_{n+1} : 1 0 1 0 ...

(* Toggle state
* Race Around cond^n
* O/P change cont.
*)

* Characteristic Table

Q_n	J	K	Q_{n+1}.
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1. ($\overline{Q_n}$)
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1.	0 ( )

* Excitation Table :

I/Ps		O/Ps	
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1.	X	0

DON'T CARE

* Conversion

* → Conversion of SR- FF to JK FF ·

1 → Identify avail and req. FF
2 → Make characteristic table for req FF
3 → Make excitation table for Available FF
4 → ~~MR~~ Write Boolean expression for
5 → Draw the circuit.

1 → SR– Available
JK - Required.

2 → Characteristic table for req. i.e JK FF.

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0.
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

3 → Excitation table for avail. ie SR. FF

I/Ps		O/Ps	
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

4 → 4 SE PEHLE
3rd table

Table for obtaining boolean expression

FIND

Q_n	J	K	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

4 → K-Map for S

Q_n \ JK	00	01	11	10
0	0	0	1	1
1	X	0	0	X

$S = \overline{Q_n} J$

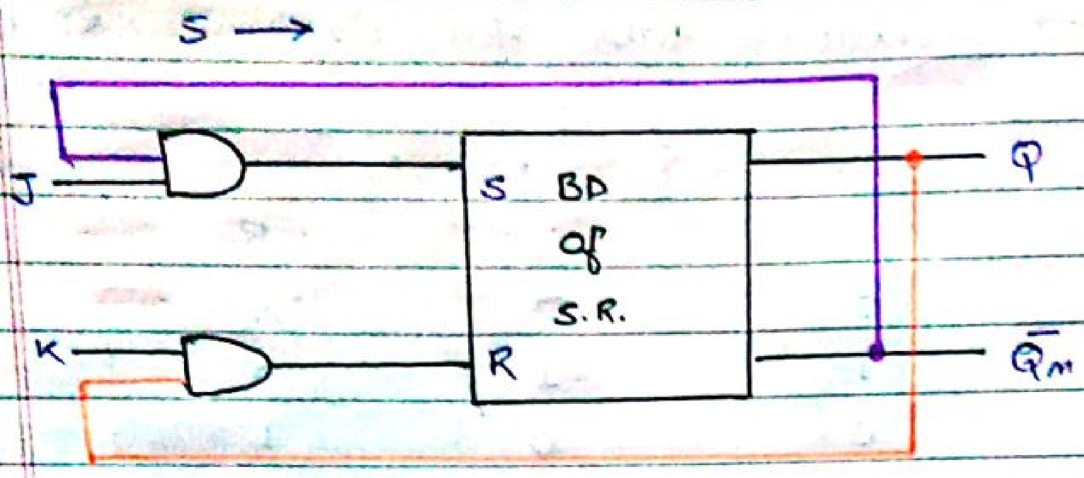
IF X then make group of 4

KMap for R

SAME HERE

Q_n \ JK	00	01	11	10
0	X	X	0	0
1	0	1	1	0

$R = Q_n K.$



Qus Convert JK-FF to D-flip flop.

1 → JK - Available
D.FF — Required

2 → Characteristic Table for req. i.e DFF.

Q	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1.

$\boxed{Q_{n+1} = D}$

3 → Excitation table for avail. i.e JK

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

3rd table

Q_n	D	Q_{n+1}	J	K.
0	0	0	0	X
0	1	1	1	X
1	0	0	X	1
1	1	1	X	0

K Map for J

Q_n \ D	0	1
0	0	1
1	X	X

TREATED AS 1

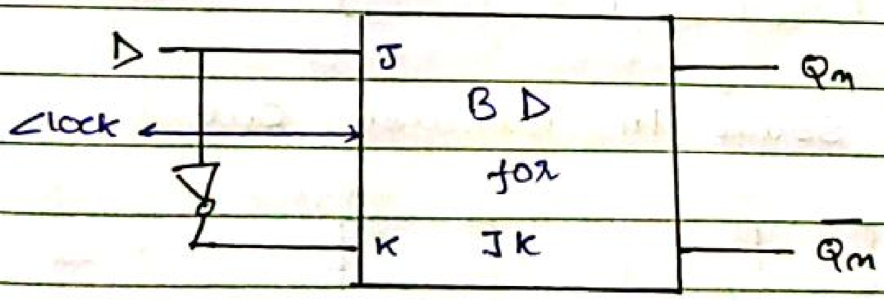
J = D

K Map for K.

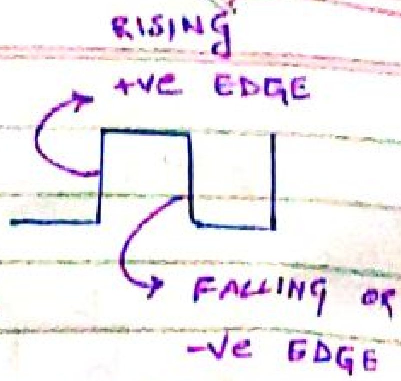
Q_n \ D	0	1
0	X	X
1.	~~X~~1	0

SAME HERE

$K = \bar{D}$



* Registers :-



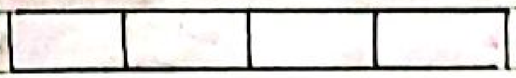
→ Made up of n flip flops
→ Storage device
→ Capacity n bits.

* Shift Register

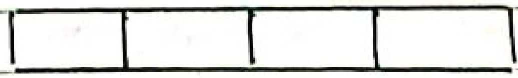
→ Able to move data

Types

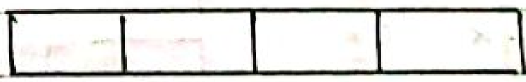
i) SISO : Serial In Serial Out.



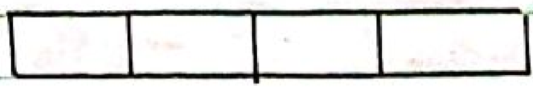
ii) PIPO : Parallel In Parallel Out



iii) SIPO : Serial In Parallel Out



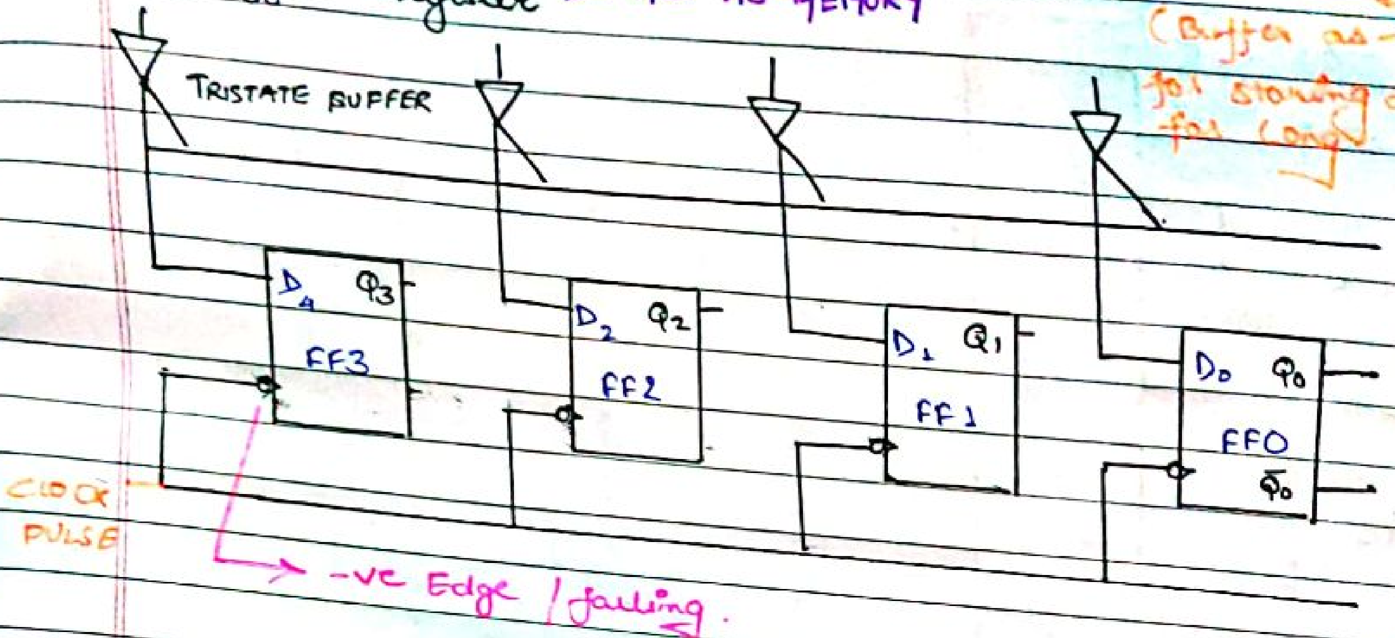
iv) PISO : Parallel In Serial Out



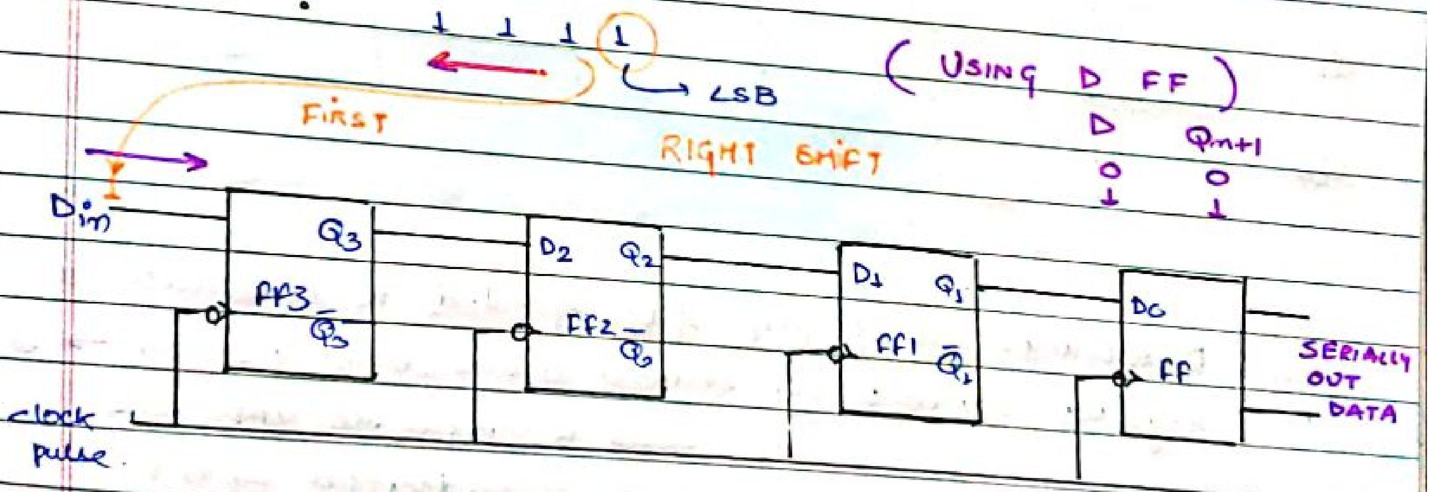
Data stored until next pulse in flip flop either rise or fall

Buffer Register → ACT AS MEMORY

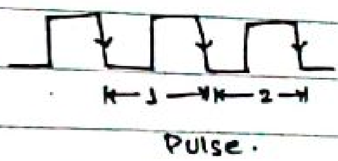
Add LOAD (Buffer as) for storing data for long



SISO :

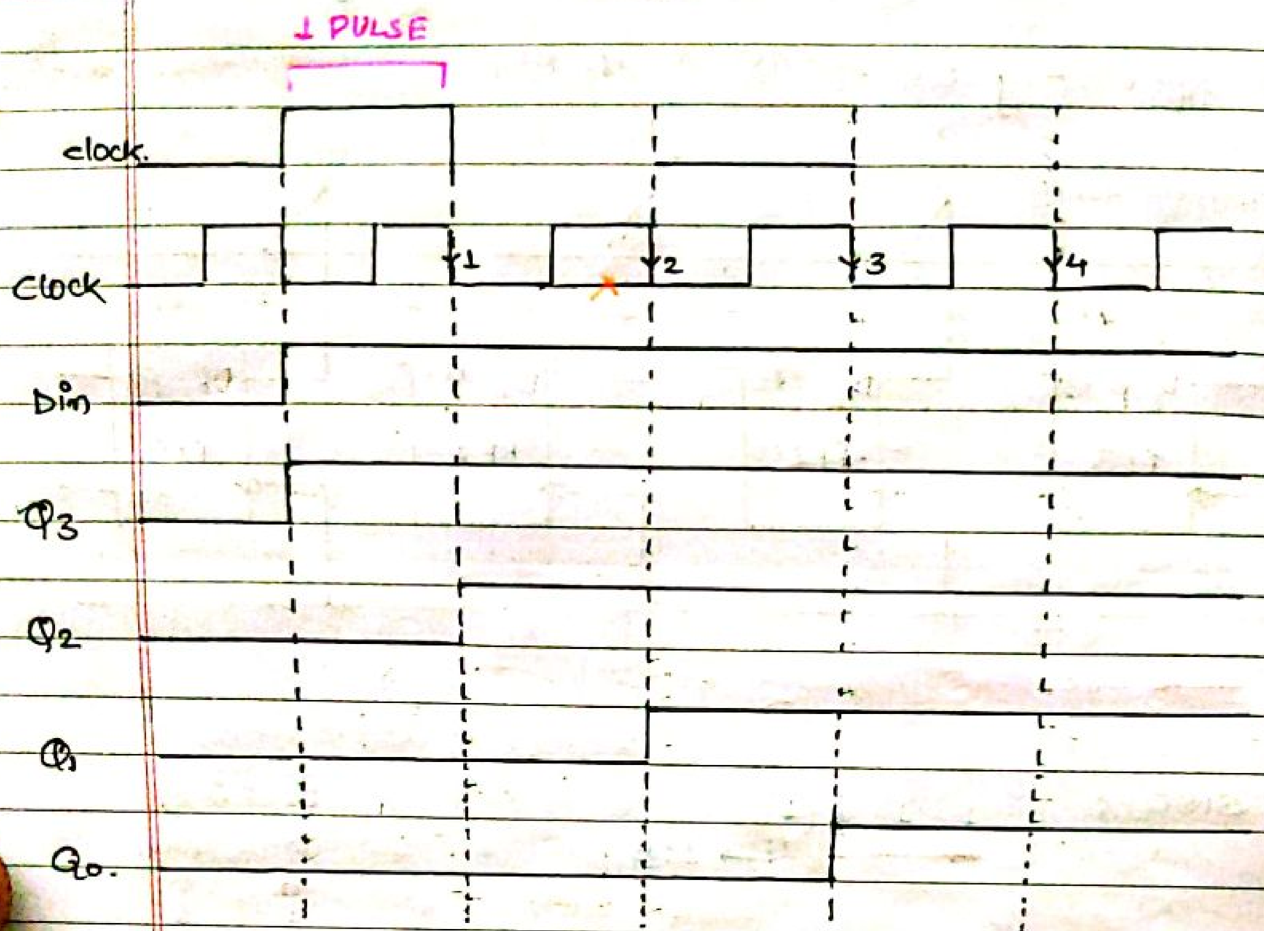


	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1.



Inputs

→ Q_3 AS INPUT FOR NEXT FF



Ques Determine no of flip flops needed to construct a register capable of storing 6 bits binary data → 6

→ Decimal no upto 32

→ Hexadecimal upto F

→ Octal upto 10 → 8

Decimal upto 32

↳ 6 bits of binary no is req to ~~cor~~ store 32 in binary

i.e 6 ffs is req.

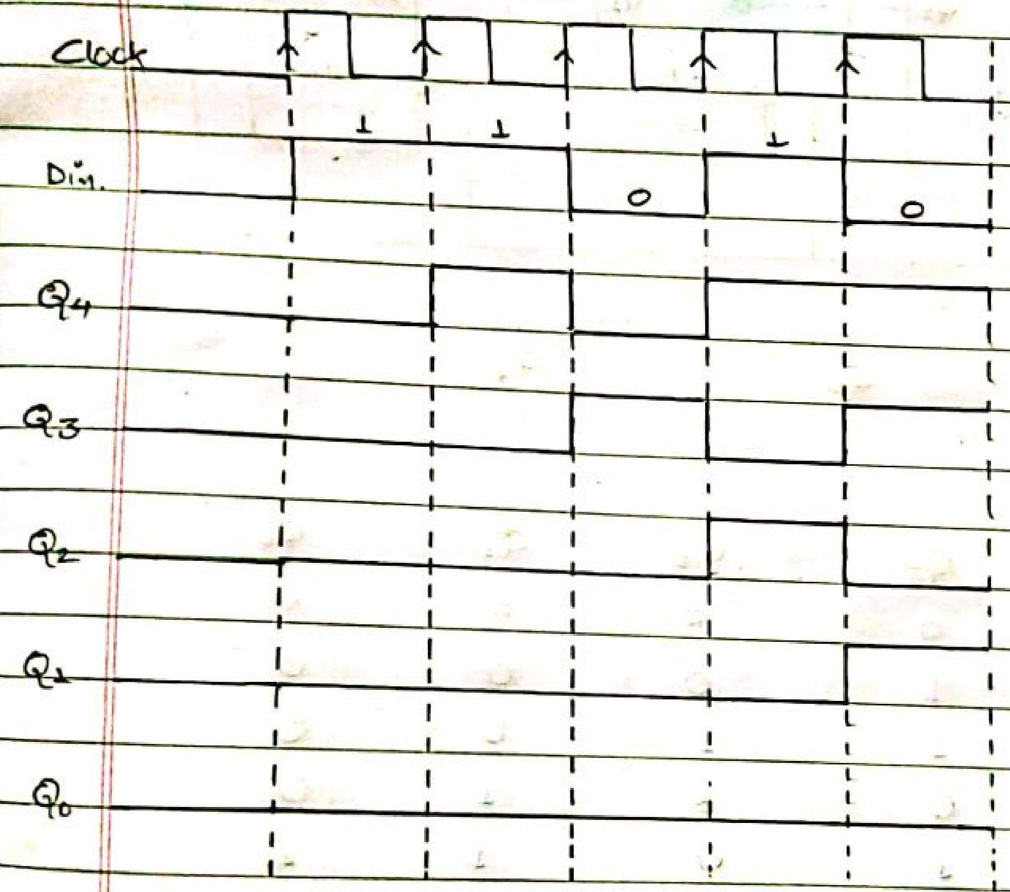
Hexa upto F (15) = 4 (1111)

= 4 ffs req.

5 FF TO BE USED

Ques. Show that status of 5 bit reg. for specified data input clocked pulse. Assume that reg. is initially clear.

I/P: 11010 (SISO)



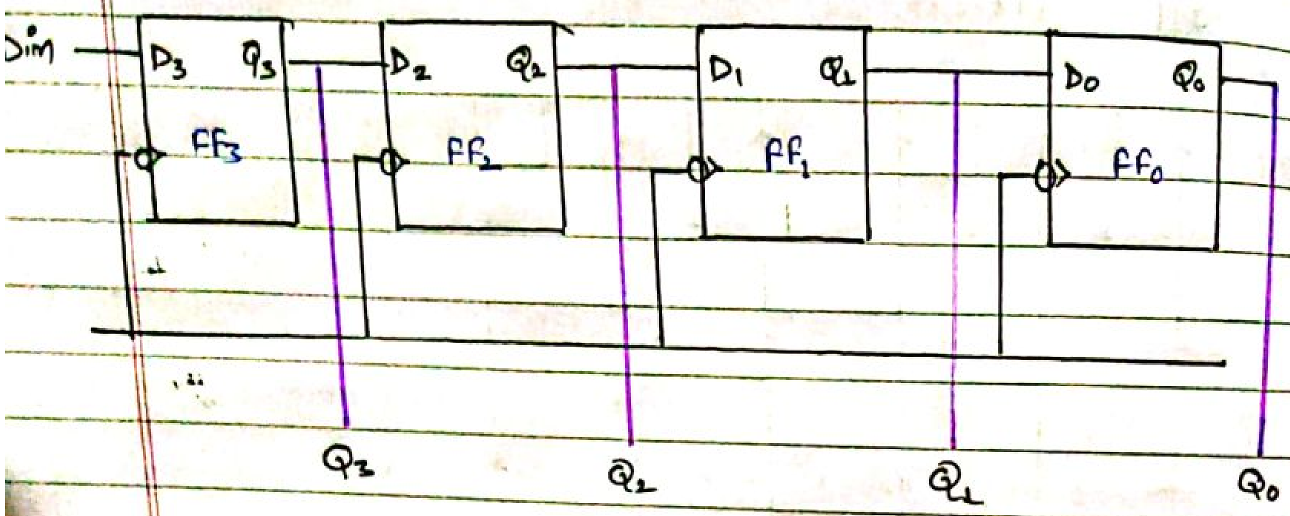
Clock	Q_4	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0	0
1	0	0	0	0	0
2	1	0	0	0	0
3	0	1	0	0	0
4	1	0	1	0	0
5	1	1	0	1	0

INPUT

Get 1 put AT A TIME

I.E. AFTER EVERY CLOCK WE GET O/P IN RESP Q.

SIPO :

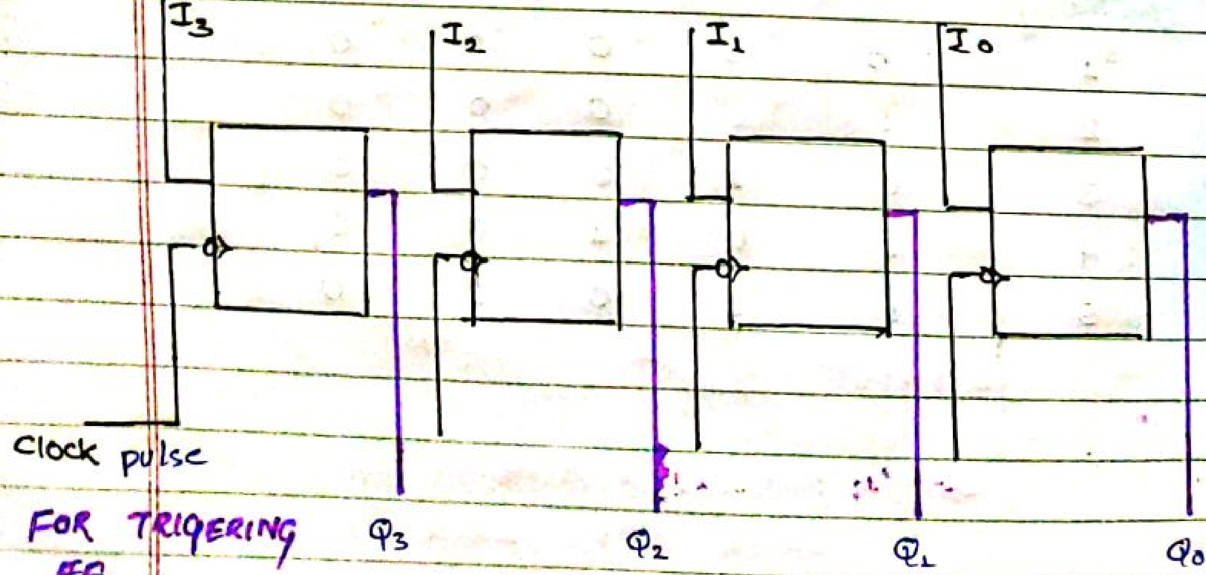


1011

	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	0	1	1	0
4	1	0	1	1.

WE GET ALL BIT SIMULTANEOUSLY AFTER 4th CLOCK.

PIPO :

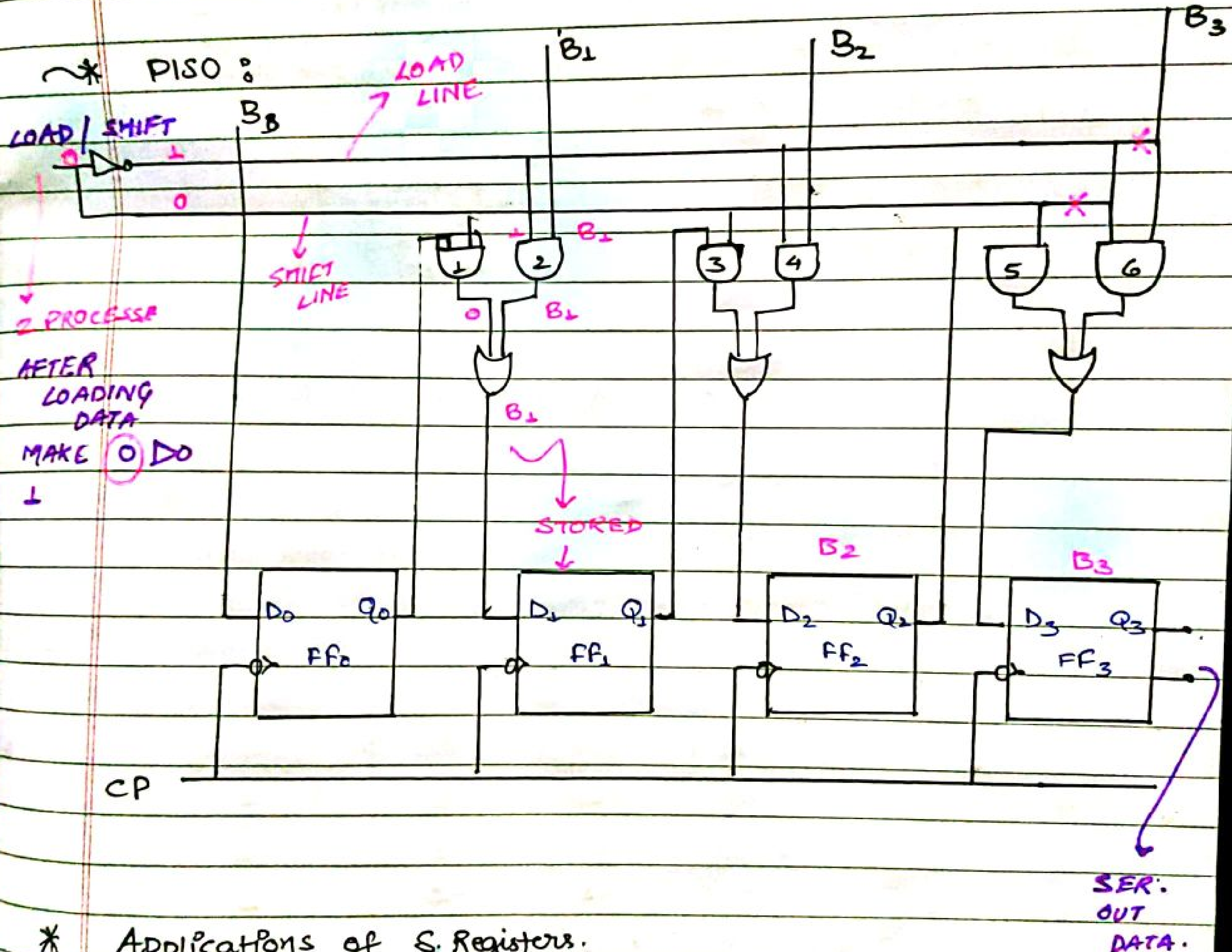


FOR TRIGERING FF.

DOESN'T REQ CLOCKED PULSE (ONLY 1 CLOCK PULSE IS ENOUGH)

INPUT				OUT_PUT			
I_3	I_2	I_1	I_0	Q_3	Q_2	Q_1	Q_0
1	1	0	0	1	1	0	0

* PISO :

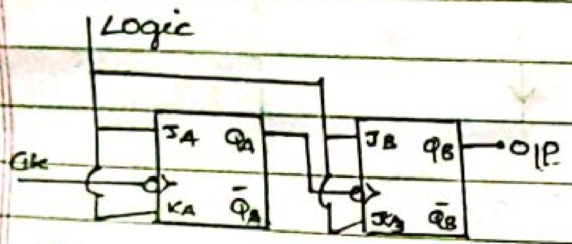


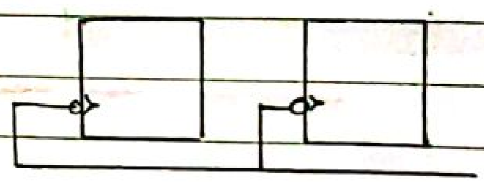
* Applications of S. Registers.

UNIT 4:

0 → 3 UP COUNTER 3 → 0 DOWN (2FF) $2^2 = 4$

Asynchronas Counter | Synchroyous Counter





0,

→ Output of 1st is clk for 2nd FF...

→ Circuit simple for more FF.

→ FF are not clocked simultayeously

→ Delay is more

Independent clk for each FF.

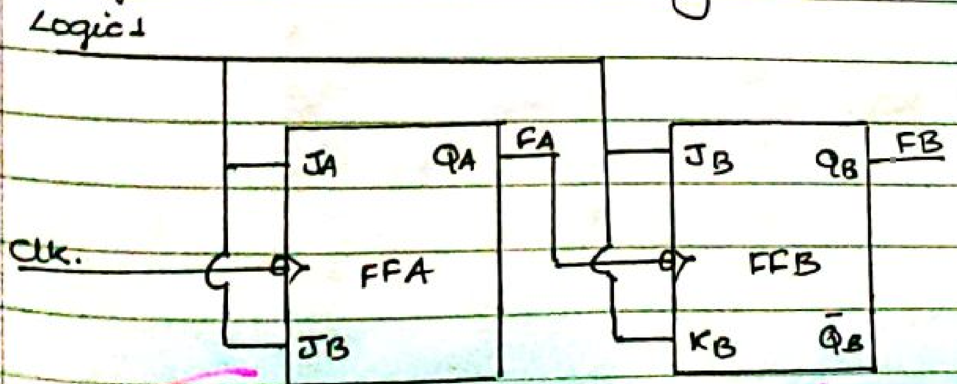
S, R, clk, J, K, D and T ⇒ SYNCRONOUS

Present, clear ⇒ ASYNCRONOUS

→ Independent of clock, IP and change O/P of F.F.

* Asynchroyous counter using J and K (ONLY → PROVIDE TOGGLE CONDN).

AT J=K=1

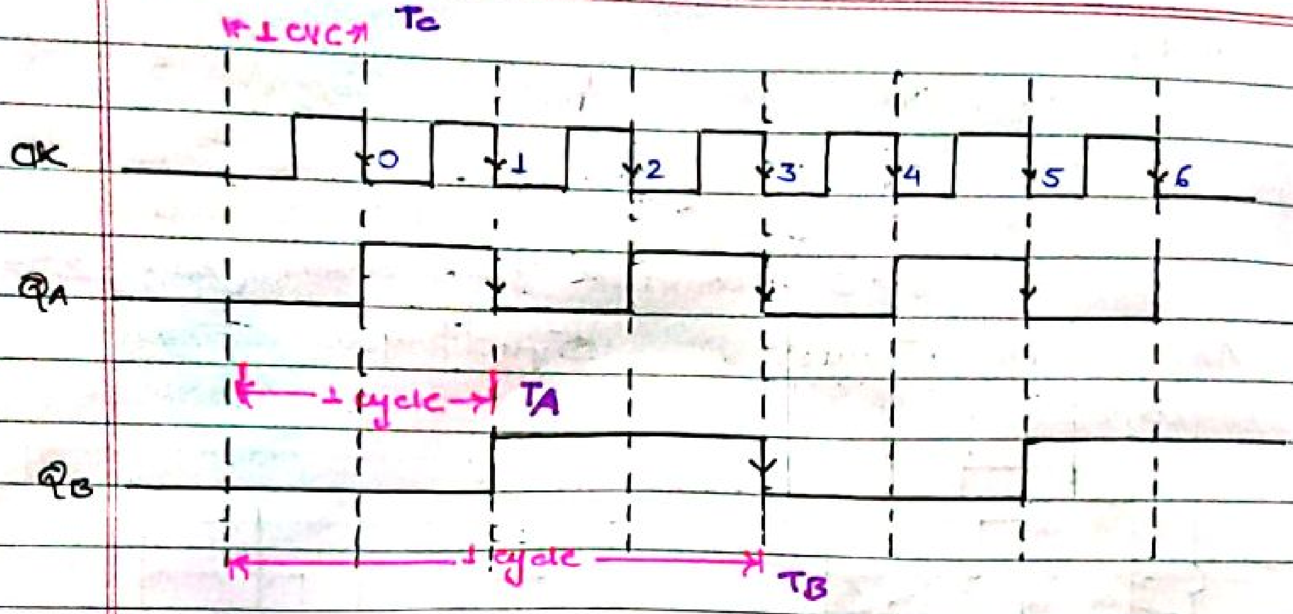


↳ ACT AS LSB

↳ LAST AS MSB

* Max count = $2^N - 1$

Eg $2^2 = 4 - 1 = 3$ $(0 \rightarrow 3)$



freq = 1/Time Period

$\therefore \boxed{f_B = \frac{f_C}{4}}$ (Divide by 4 freq.)

↓

[DEPENDS ON NO. OF FF. $2^2 = 4$]

aka MOD 4 COUNTER.

Clk	Q_B	Q_A	Decimal
0	0	0	0
(↓) 1st Pulse	0	1	1
(↓) 2nd.	1	0	2
(↓) 3rd	1	1	3
(↓) 4th	0	0	0

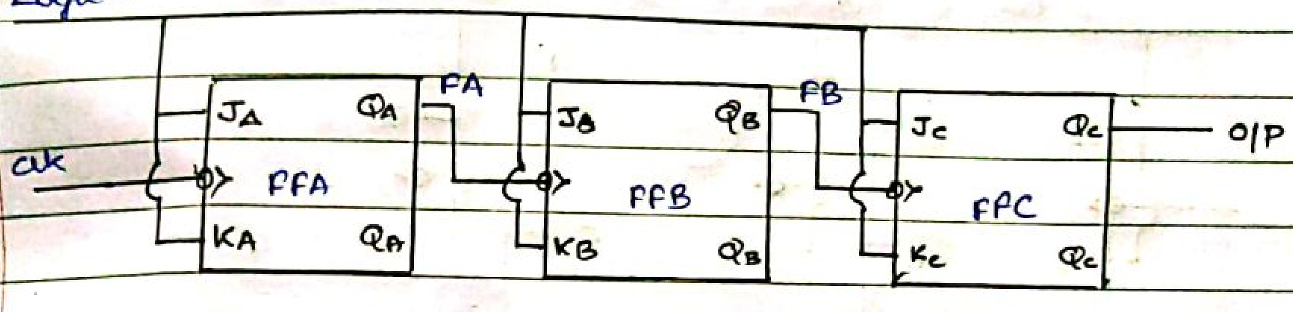
↑sing ⇒ UP COUNTER

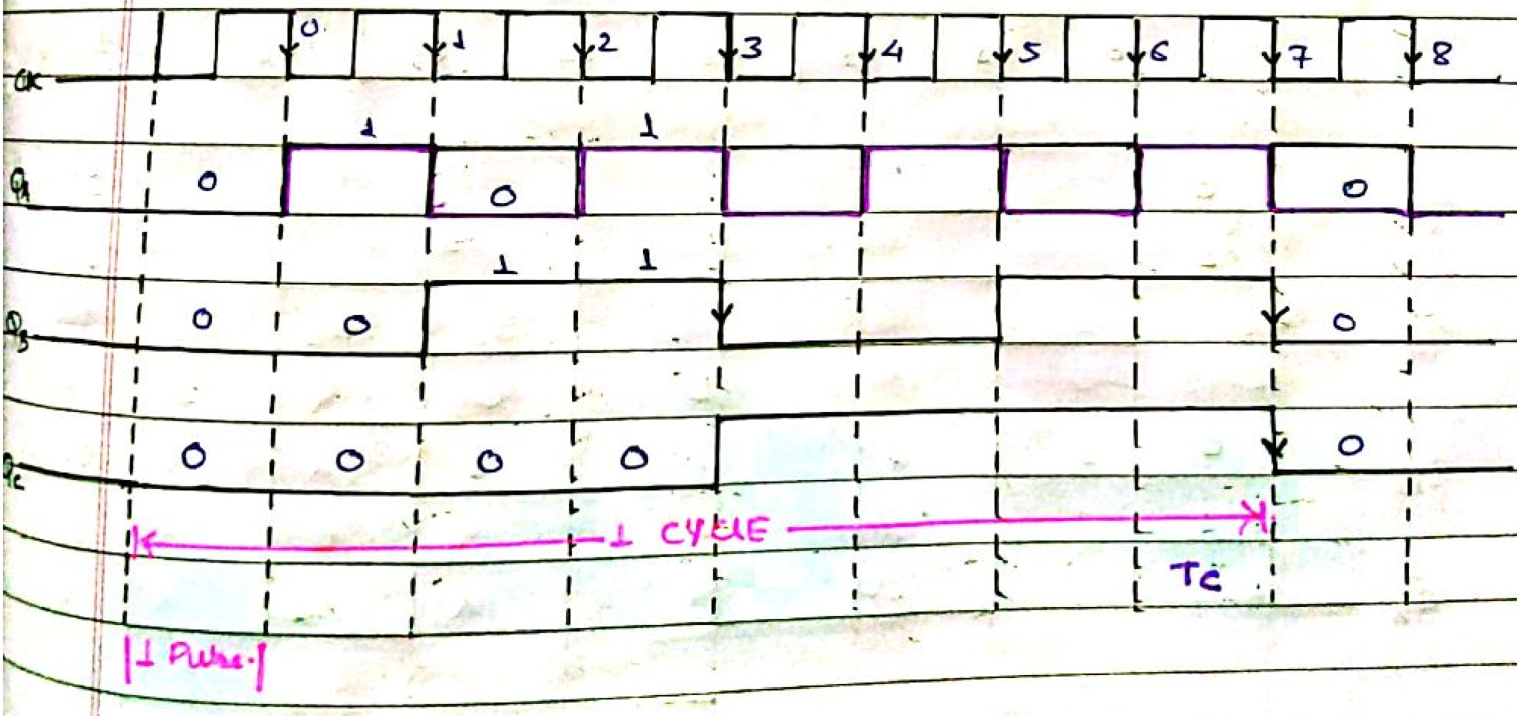
$0 \rightarrow 3$

Scanned by CamScanner

Qu Design 3 bit Asyn. UP counter.

Logic 1



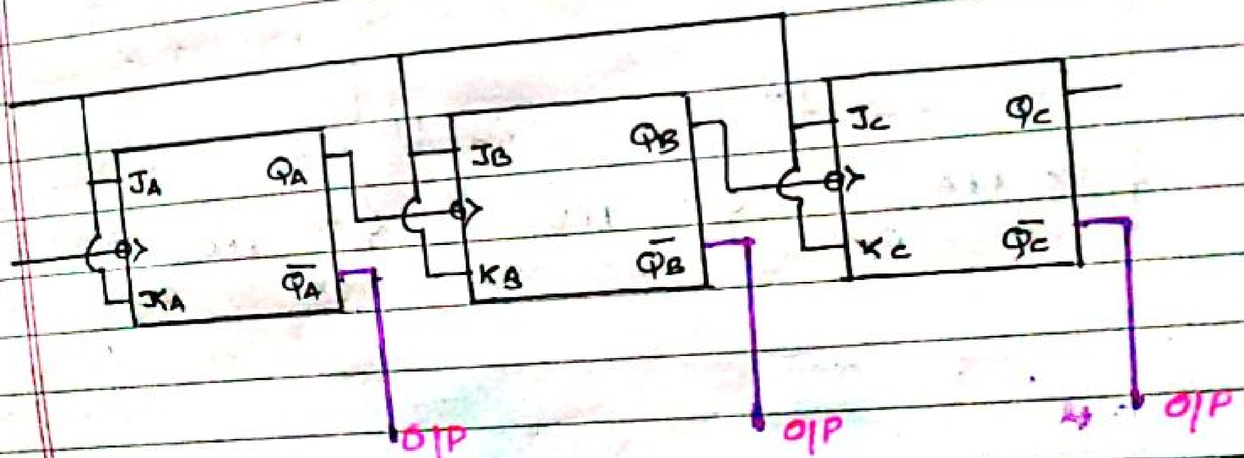


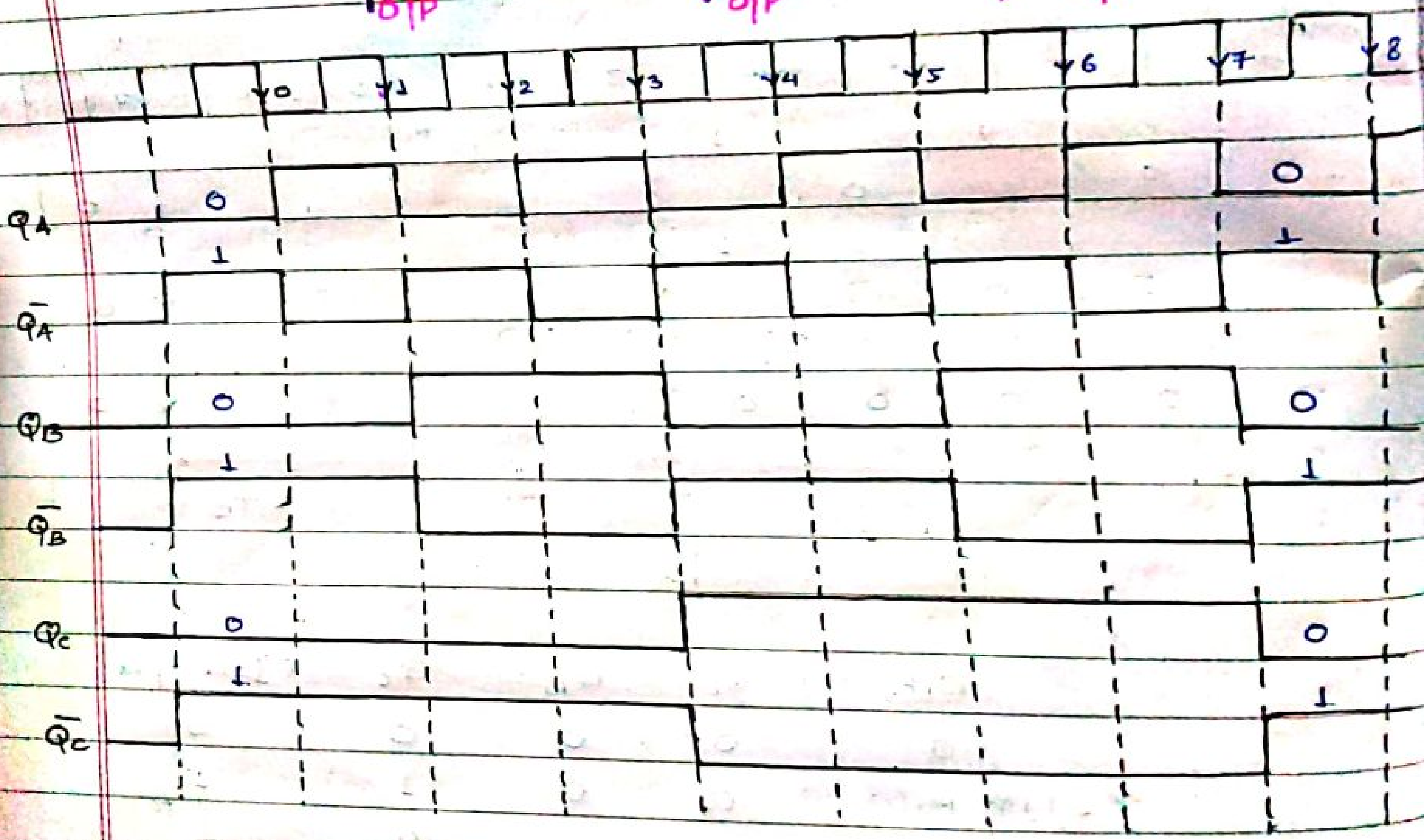
Clk	Qc	QB	Qc	Decimal
0	0	0	0	0
(↓) 1st Pulse	0	0	1	1
2nd	0	1	0	2
3rd	0	1	1	3
4th	1	0	0	4
5th	1	0	1	5
6th	1	1	0	6
7th	1	1	1	7
8th	0	0	0	0

UP COUNTER (↑ing)

0 → 7

Que Design 3 bit Async. down counter.





Scanned by CamScanner

Clk	Q_C	Q_B	Q_A	$\bar{Q}_C$	$\bar{Q}_B$	$\bar{Q}_A$	Decimal
0th	0	0	0	1	1	1	7
1st	0	0	1	1	1	0	6
2nd	0	1	0	1	0	1	5
3rd	0	1	1	1	0	0	4
4th	0	0	0	0	1	1	3
5th	1	0	1	0	1	0	2
6th	1	1	0	0	0	1	1
7th	1	1	1	0	0	0	0
8th	0	0	0	1	1	1	7.

↓ sing DOWN COUNTER

(7→0)

* Designing Sync. FF.

→ Decide no. of FFs.
→ Excitation table of FF.
→ [State diag.] and [circuit excitⁿ table]
→ Obtain simplified eqn using K-MAP
→ Draw Logic diagram.

Que Design 2 bit Sync. UP counter using JK FF.

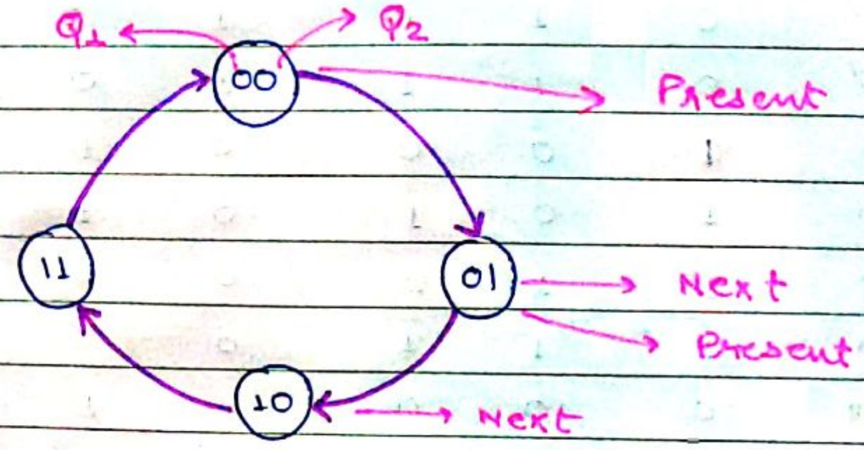
→ As it is a 2 bit FF we need 2 JK FF.

→ Excitation table for JK

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

→ State diagram and circuit excit^{n} table.

→ No of state = 2^{n} = 2^2 = 4 states



Circuit excitation table

2nd FF

1st FF

Present State		Next State		I/P			
Q_1	Q_2	Q_1^*	Q_2^*	J_1	K_1	J_2	K_2
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	~~0~~1	1	X	0	1	X
1	1	0	0	X	~~0~~1	X	1

PREV (PRESENT)

NEXT

FROM ~~EXCITATION TABLE~~ STATE DIAG

K-Map for J_1.

$Q_1 \backslash Q_2$	0	1
0	0	$\boxed{1}$
1	X	$\boxed{X}$

BECAUSE GIVEN ∴

$J_1 = Q_2$

K-Map for K_1

$Q_1 \backslash Q_2$	0	1
0	X	$\boxed{X}$
1	0	$\boxed{1}$

$K_1 = Q_2$

Scanned by CamScanner

K-Map for J_2

$Q_1 \backslash Q_2$	0	1
0	1	X
1	1	X

$J_2 = 1$

K-Map for K_2

$Q_1 \backslash Q_2$	0	1
0	X	1
1	X	1

$K_2 = 1.$

→ Draw Logic diagram.