

Flips flop (cont.)

Truth Tables :

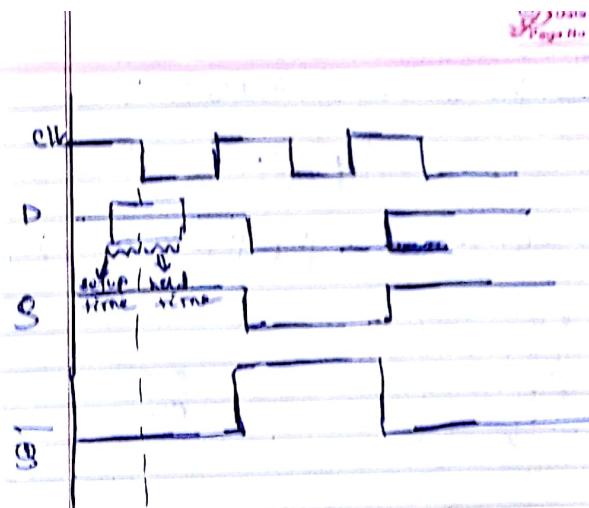
+ve edge triggered : -ve edge triggered

| clk | J | K | Q^+ | clk | J | K | Q^+ |
|-----|---|---|--------|-----|---|---|--------|
| ↓ | 0 | 0 | NC | ↑ | 0 | 0 | NC |
| ↓ | 0 | 1 | Reset | ↑ | 0 | 1 | Reset |
| ↓ | 1 | 0 | Set | ↑ | 1 | 0 | Set |
| ↓ | 1 | 1 | Toggle | ↑ | 1 | 1 | Toggle |
| 0 | X | X | NC | 0 | X | X | NC |
| 1 | X | X | NC | 1 | X | X | NC |
| ↑ | X | X | NC | ↓ | X | X | NC |

If a timing diagram is given, then write the output, which assuming any one case (say +ve pulse or -ve pulse etc)

For D flip flop :

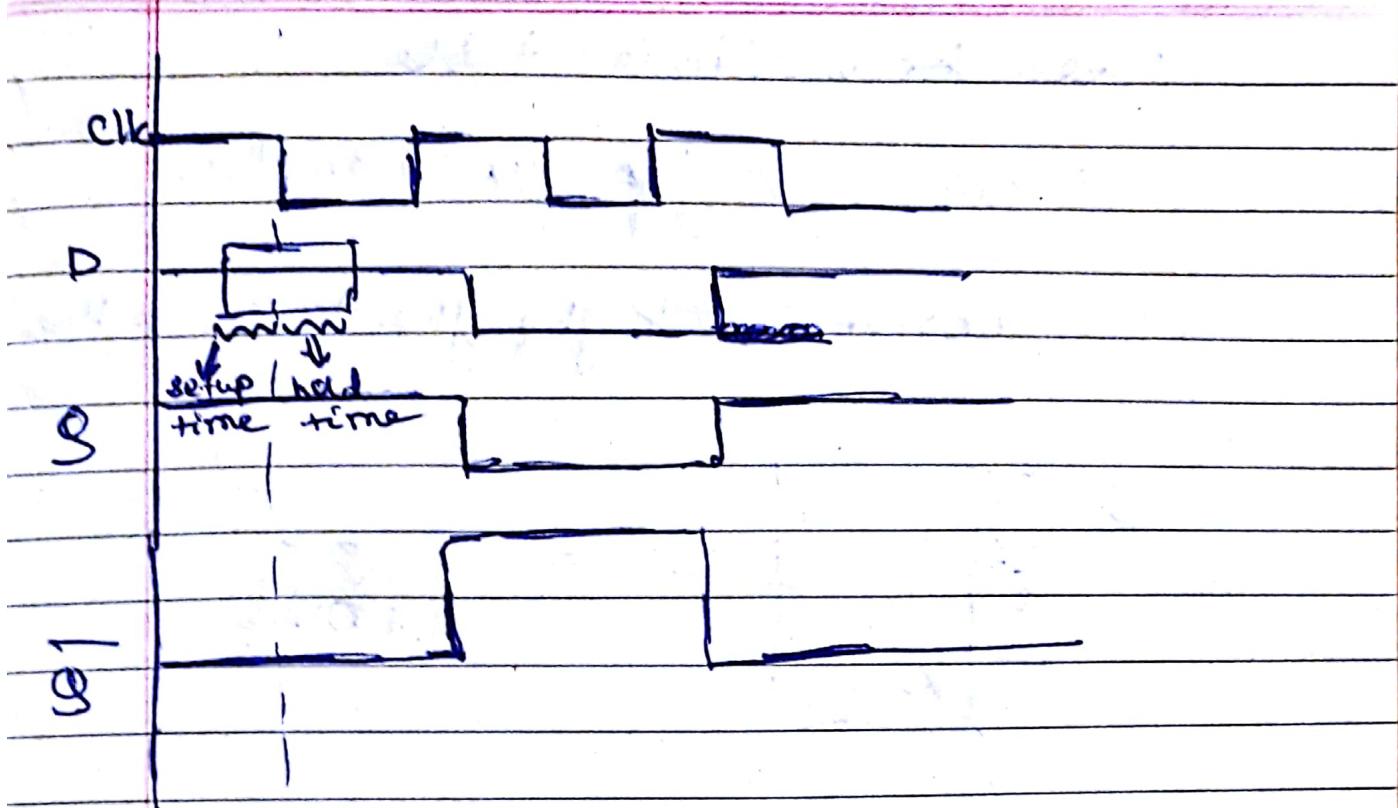
Assuming +ve pulse & zero propagation delay.



D (here we write D)
Minimum time the signal must be held fixed before the latching action.

→ Setup time.
Minimum time the D signal must be held fixed after the latching action
→ Hold time

Note: There is some minimum length of input required to compare to width of clock signal, to get desired output → This is called minimum pulse width.



D (when we write D)
 Minimum time the signal must be held fixed before the latching action.

↳ setup time.

Minimum time the D signal must be held fixed after the latching action

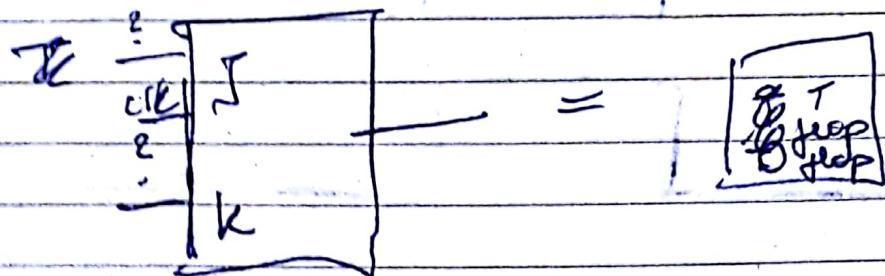
↳ hold time

Note: There is some non-length of input required for comparison to width of clock signal, to get desired output
 → This is called minimum pulse width,

Need for excitation table:

To convert 1 flip flop to another

eg: Convert JK flip flop to T flip flop.



func-table of T flip flop:

| T | Q | Q ⁺ | J | K |
|---|---|----------------|---|---|
| 0 | 0 | 0 | 0 | X |
| 0 | 1 | 1 | X | 0 |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | X | 1 |

func-table
of T flip flop

excitation table
of JK flip flop

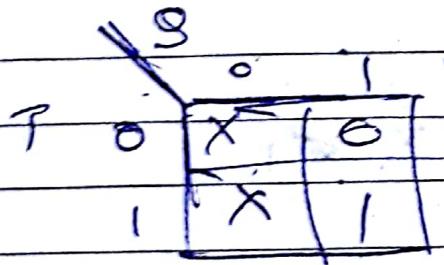
To get J & K:

~~consider~~ consider T with Q for

| T | Q ₀ | Q ₁ | (J) J |
|---|----------------|----------------|-------|
| 0 | 0 | 1 | X |
| 1 | 1 | X | 1 |

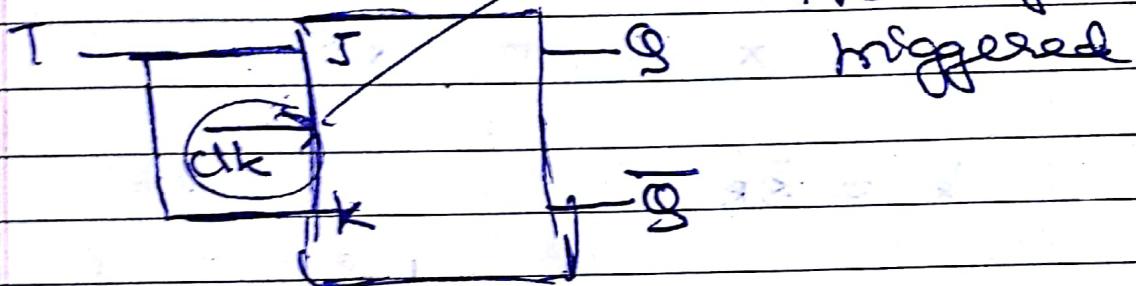
$J = T$

Consider T with S for (ii) k



$k = T$

\therefore Conversion \rightarrow implies
+ve edge triggered



Convert JK to SR:

(for forbidden state - should be considered)

e.g:

| S | R | Q | Q^+ | J | K |
|-----|-----|-----|-------|-----|-----|
| 0 | 0 | 0 | 0 | 0 | X |
| 0 | 0 | 1 | 1 | X | 0 |
| 0 | 1 | 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 | X | 1 |
| 1 | 0 | 0 | 1 | 1 | X |
| 1 | 0 | 1 | 1 | X | 0 |
| 1 | 1 | 0 | 1 | 1 | X |
| 1 | 1 | 1 | 1 | X | 0 |

Note: only while making the excitation table, we don't consider f.s. All other causes we have to.

$\therefore J:$

| S \ RQ | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | X | X | 0 |
| 1 | 1 | X | X | X |

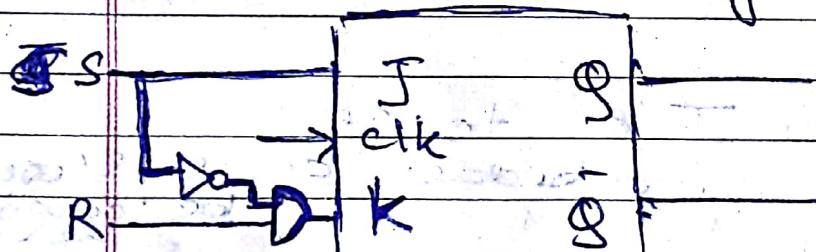
$$J = S$$

$K:$

| S \ RQ | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | X | 0 | 1 | X |
| 1 | X | 0 | 0 | X |

$$K = \bar{S}R$$

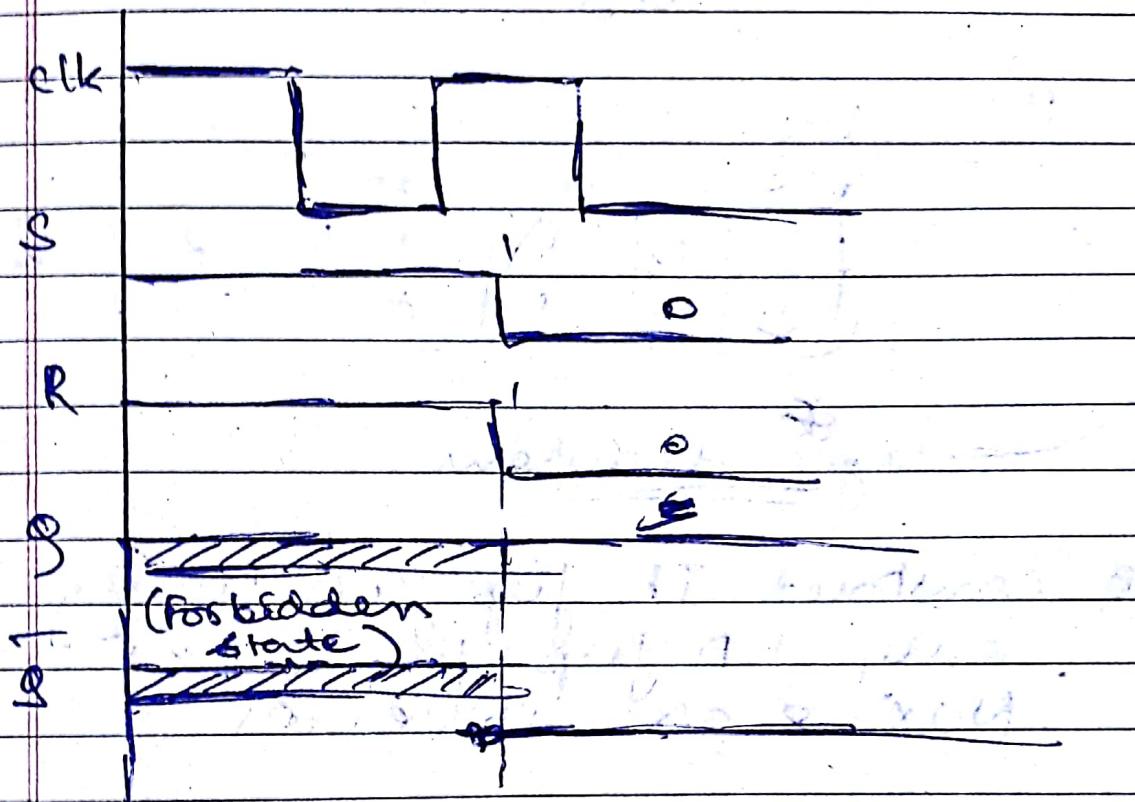
Logic diagram



Convert JK to SR

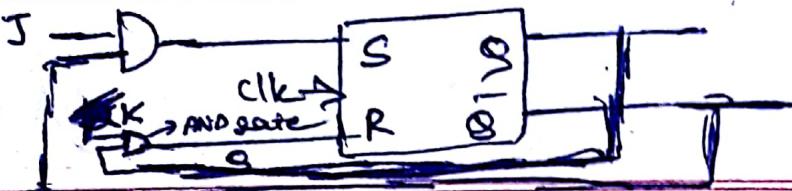
Note: For forming diagram

* I'm an SR flip flop!
(Assuming +ve pulse triggered)



Convert SR to JK

| J | K | Q_1 | Q_0 | S | R | |
|---|---|-------|-------|---|---|--|
| 0 | 0 | 0 | 0 | 0 | x | |
| 0 | 0 | 1 | 1 | x | 0 | |
| 0 | 1 | 0 | 0 | 0 | x | |
| 0 | 1 | 1 | 0 | 0 | 1 | |
| 1 | 0 | 0 | 1 | 1 | 0 | |
| 1 | 0 | 1 | 1 | x | 0 | |
| 1 | 1 | 0 | 1 | 1 | 0 | |
| 1 | 1 | 1 | 0 | 0 | 1 | |



S:

| J | 00 | 01 | 11 | 10 |
|---|----|----|----|----|
| 0 | 0 | X | 0 | 0 |
| 1 | 1 | X | 0 | 1 |

$$S = J\bar{Q}$$

R:

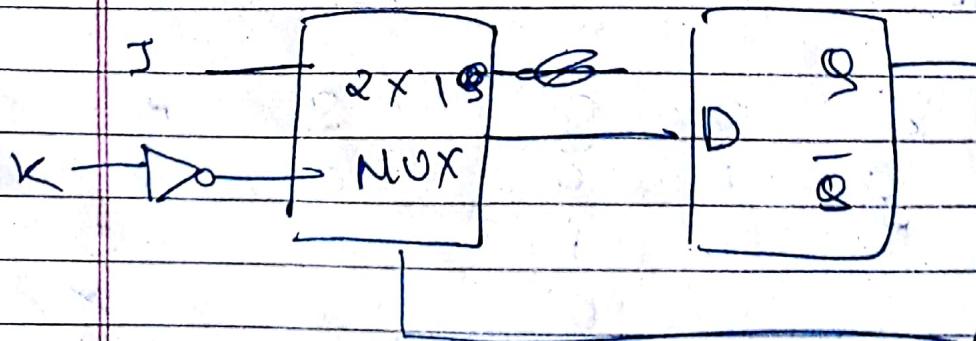
| J | 00 | 01 | 11 | 10 |
|---|----|----|-----|----|
| 0 | X | 0 | (0) | X |
| 1 | 0 | 0 | (1) | 0 |

$$R = K\bar{Q}$$

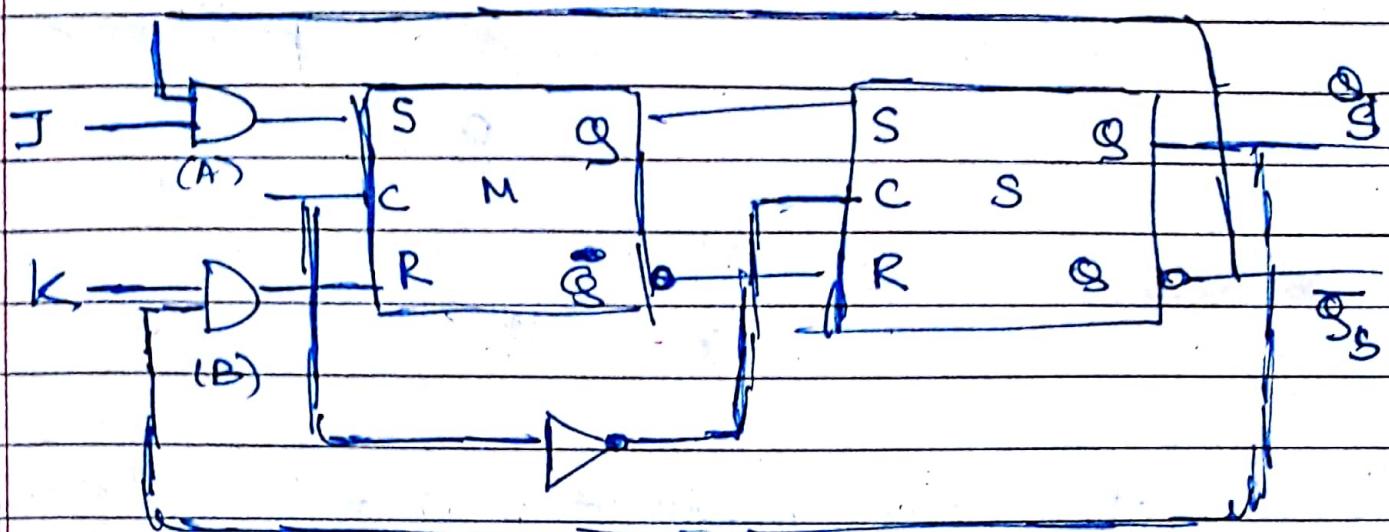
Logic diagram

Q. construct JK flip flop using only 1 D flip flop, 2×1 NUX & an inverter.

Ans.



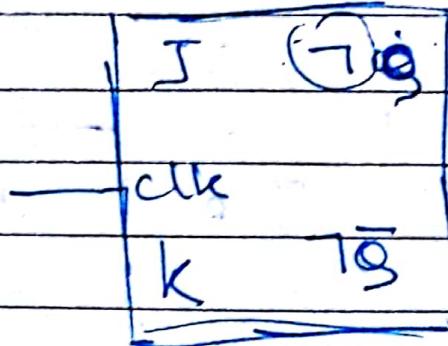
Master - Slave Flip flop:



Waveform for explanation

Logic Symbol for Master-Slave Jk flip flop

→ implies delay



Effectively, ~~Master does what slave does~~ whatever the master does, but with $\frac{1}{2}$ pulse width delay (assuming no propagation delay)

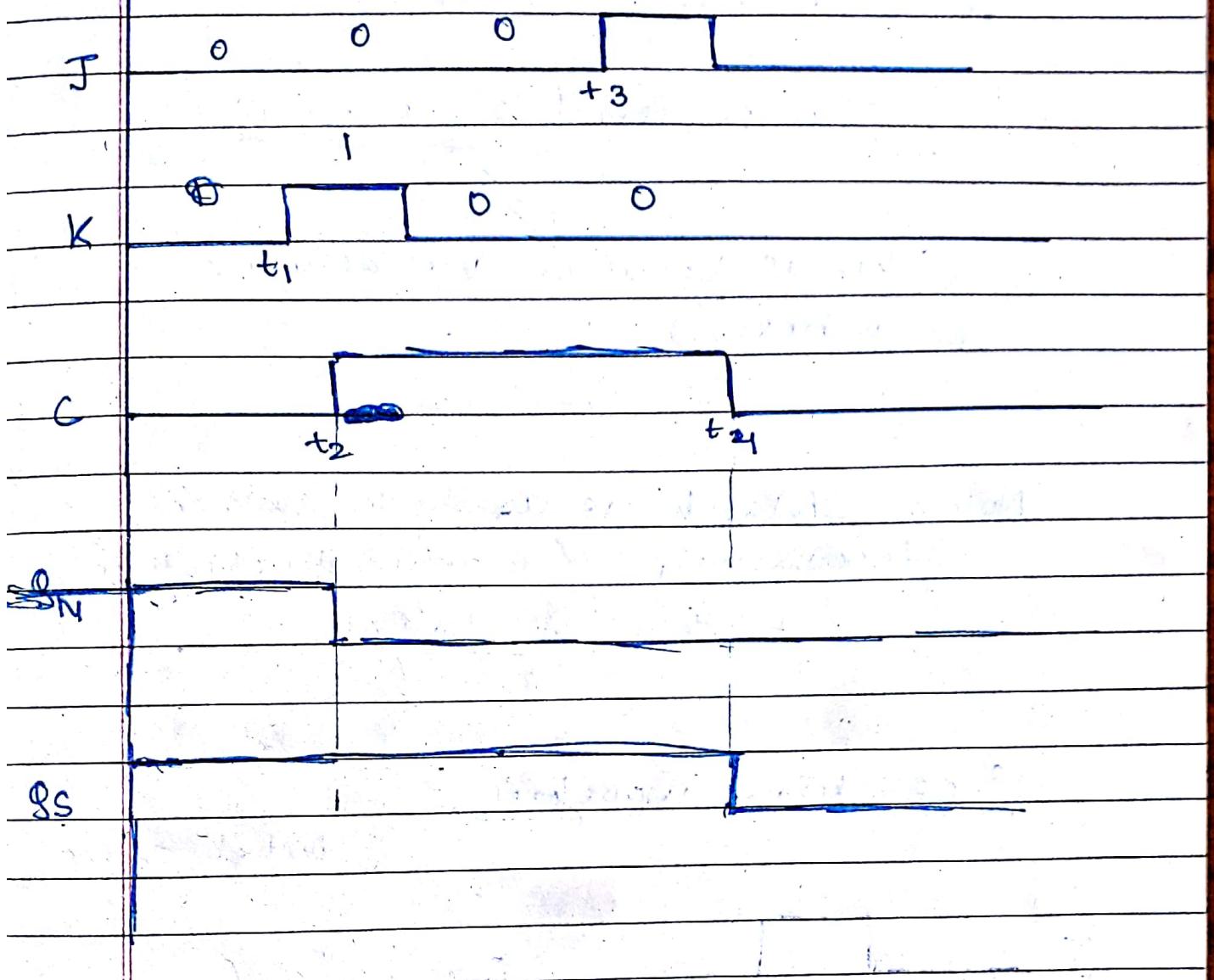
Master slave overcomes the race condition with delayed output
⇒ Toggles only 1

1's catching / 0's catching : Disadv. of NS flip flop.

Zero's Catching

Now, consider the waveform shown, get $Q_N = S Q_S$

Assume Q_S (initial) = 0 |



At t_1 , the K input goes to 1. Since Q_S is 1 at t_2 , when the clock goes high, the output of and gate B (lower one) goes to 0, so the master resets (0). Note that Q_S is still at logic 1.

At t_3 , the J input goes high. Since Q_S is still zero, the output of AND gate A (upper) is 0, so the $J = 1, K = 0$ input goes unrecognised.

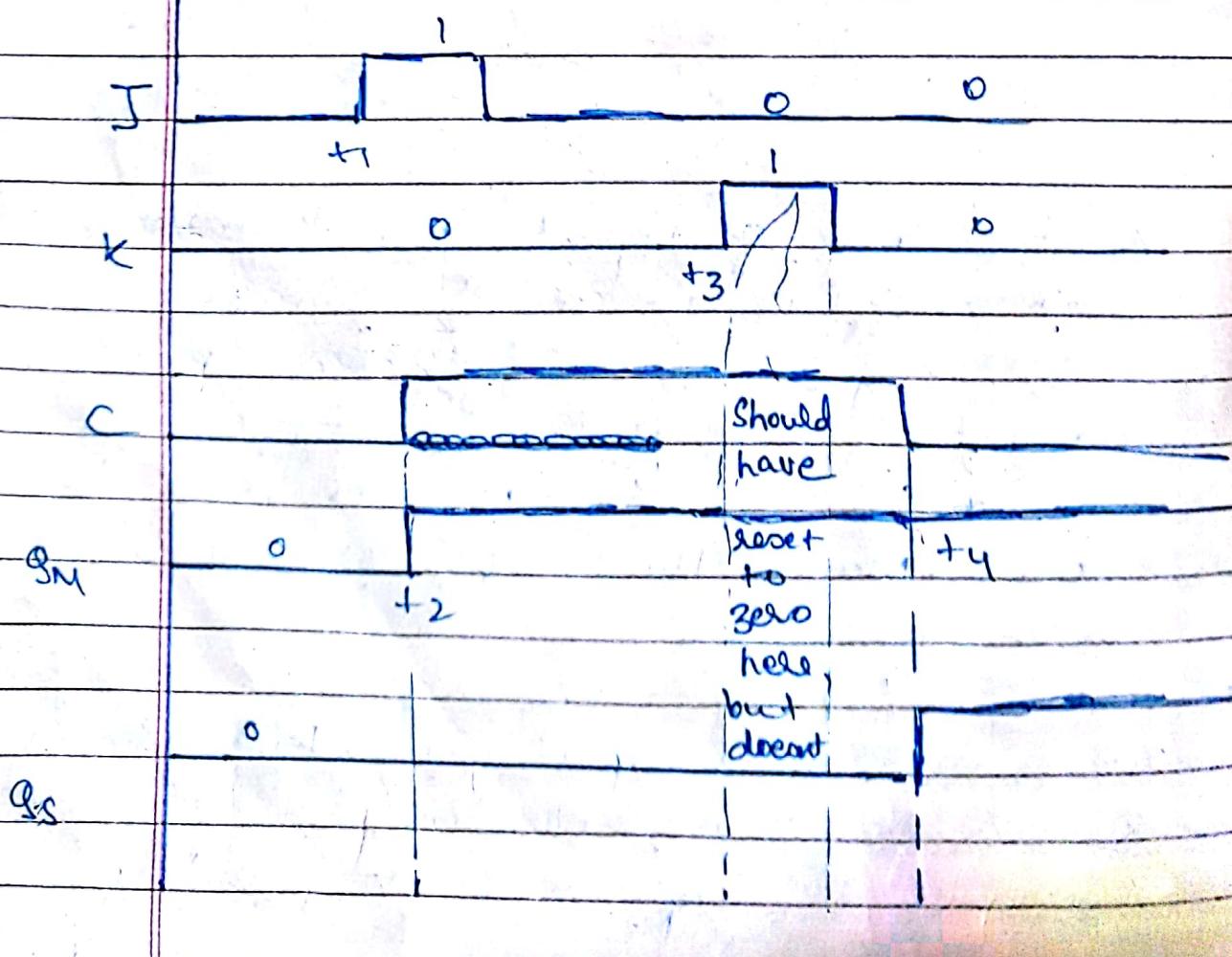
The slave resets at the falling edge of the clock at t₄.
This is called zero's catching.

(This is because we assume Q_S = 1. here)

Note: SR latch is used in switch debouncing (to avoid fluctuation)
↳ How? See online

1st catching waveform:

initial Q_S = 0



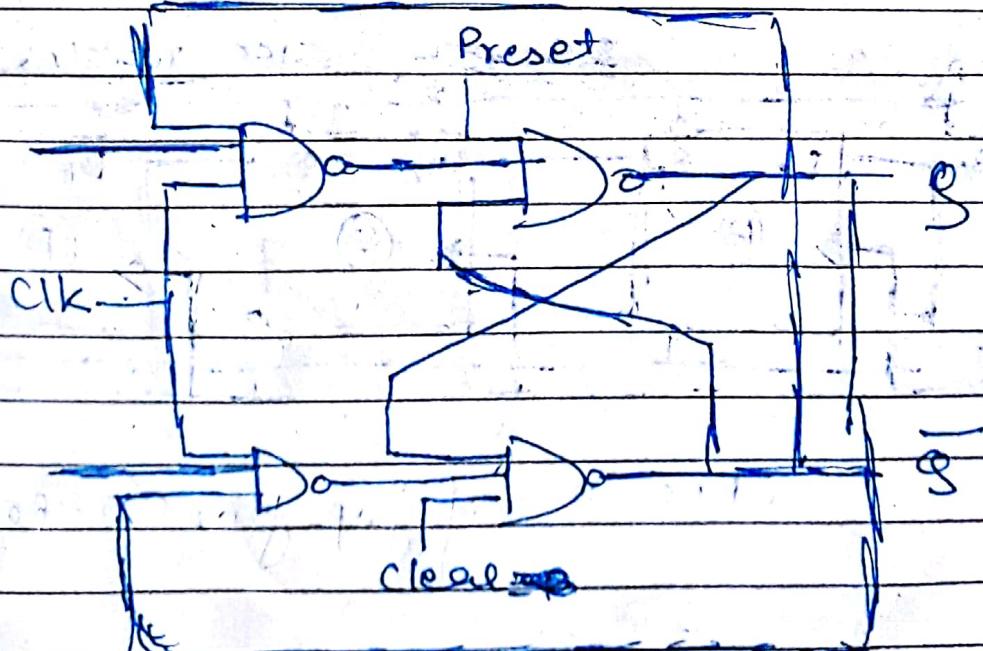
At t_1 , the J input goes to 1. Since $\overline{Q_S}$ is 1 at t_2 , when the clock goes high, the output of and gate B goes to 0. So the master sets. Note that Q_S is still at 0.

At t_3 , the K input goes high. Since $Q_S = 0$ still, too.

the $J=0, K=1$ input goes unorganized

Preset & clear (Asynchronous inputs)

↳ To either completely set or clear.



without clock \Rightarrow this becomes a latch.

Truth Table

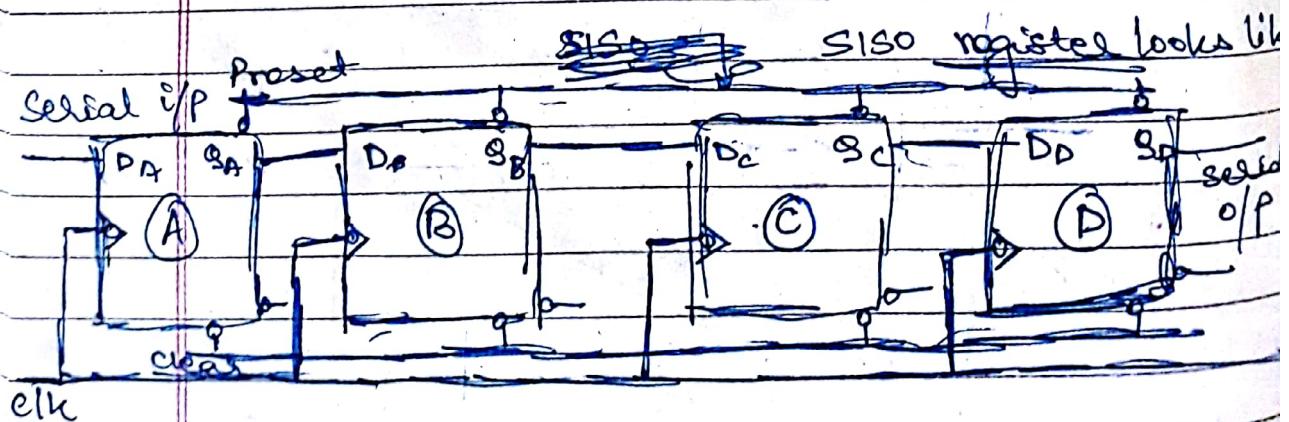
| Pre | Clear | Clock | J | K | S |
|-----|-------|-------|---|---|-----------------|
| 1 | 1 | 1 | 0 | 0 | N.C |
| 1 | 1 | 1 | 0 | 1 | Set Reset |
| 1 | 1 | 1 | 1 | 0 | Reset |
| 1 | 1 | 1 | 1 | 1 | Toggle |
| 1 | 1 | 0 | X | X | No change |
| 0 | 1 | X | X | X | 1 |
| 1 | 0 | X | X | X | 0 |
| 0 | 0 | - | - | - | forbidden state |

Why is preset & clear active low?

→ ~~So that the~~ since we are using NAND gates, by making the input low, it is possible to obtain the 1 for set or reset at the corresponding end.

(other concepts of flip flop have been)

Registers



Connection of flip flops : Registers

Here, we use D flip (one edge triggered)



Lecture 8: Registers

Types of Registers:

(i) Serial in Serial out (SISO)

(ii) I/P: serially

O/P: obtained serially

(iii) SIPO (Serial in parallel out)

I/P: serial O/P: parallel

(iv) PIPO

(v) PISO.

For the diagram on prev. page

Initially, clear all the contents

\Rightarrow set value of each

register to 0 (using CLEAR
asynchronous i/p)

\Rightarrow O/P: 0

SI Q_n Q_{n-1} Q_{n-2} Q_{n-3}

Q_n Q_{n-1} Q_{n-2} Q_{n-3}

④

clk SI Q_n Q_{n-1} Q_{n-2} Q_{n-3}

clk \rightarrow 1 \rightarrow 1 \rightarrow 0 \rightarrow 0 \rightarrow 0

Here, all flipflops, clock is applied at
the same time. (\because all flipflops are
synchronous)

The ~~Q_A~~^{exists} is i/p to \textcircled{B} , Q_B is ~~existing~~
is i/p to \textcircled{C} , Q_C to \textcircled{D} etc.

∴ The ~~of~~ table goes as follows

| CK _n | SI | Q_A | Q_B | Q_C | Q_D |
|-----------------|----|-------|-------|-------|-------|
| clk → 1 | 1 | 1 | 0 | 0 | 0 |
| clk → 2 | 1 | 1 | 1 | 0 | 0 |
| clk → 3 | 0 | 0 | 1 | 1 | 0 |

Note that, since this is serial i/p & serial o/p - only Q_D is visible as output.

$Q_A, Q_B, Q_C \rightarrow$ not visible

When endata is given in serial i/p, to see it ~~as~~ output have to wait for n ~~clocks~~^{clocks}, after i/p given, where, n is the no. of flip flops.

SISO: Similar to SISO, except,
 Q_A, Q_B, Q_C are also visible
(can be taken out)

Application for SISO:

↳ to transmit serial data into parallel data

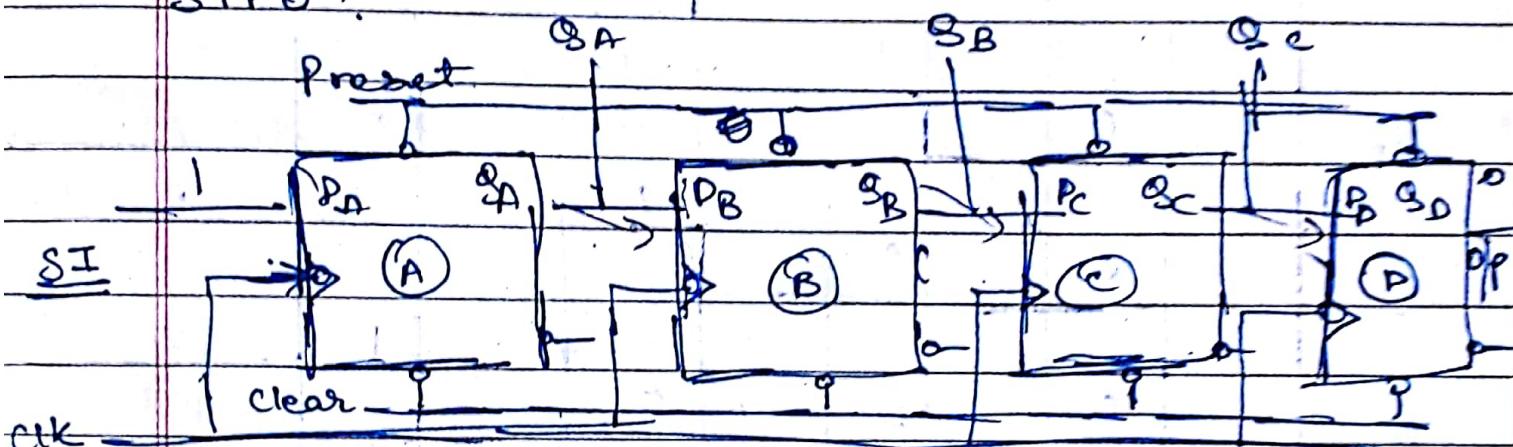
PISO:

↳ to transmit parallel data into serial data

SISO:

↳ to create a delay

SISO:



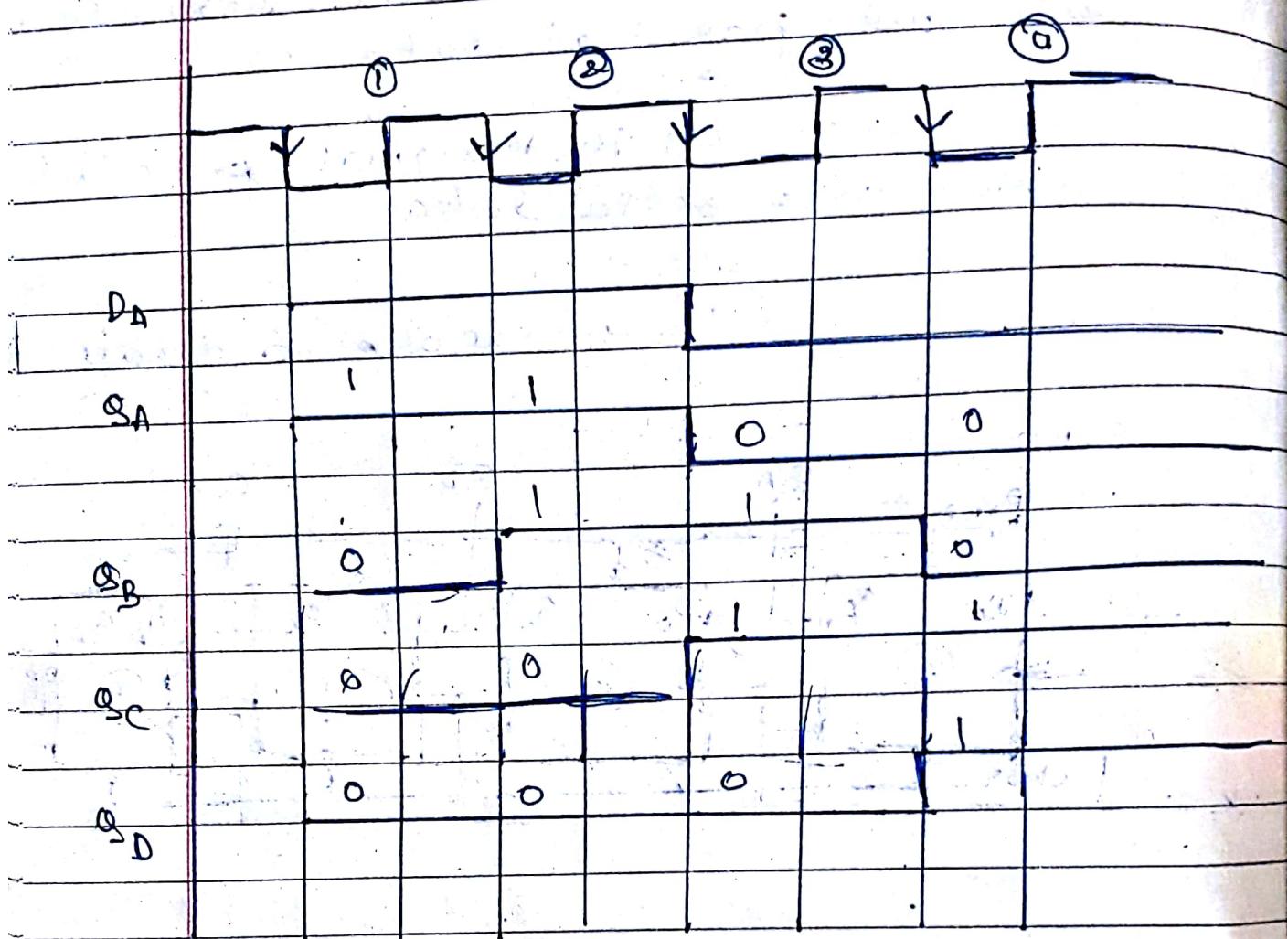
for a serial in serial out

flip :

$$\Delta t = N \times 1$$

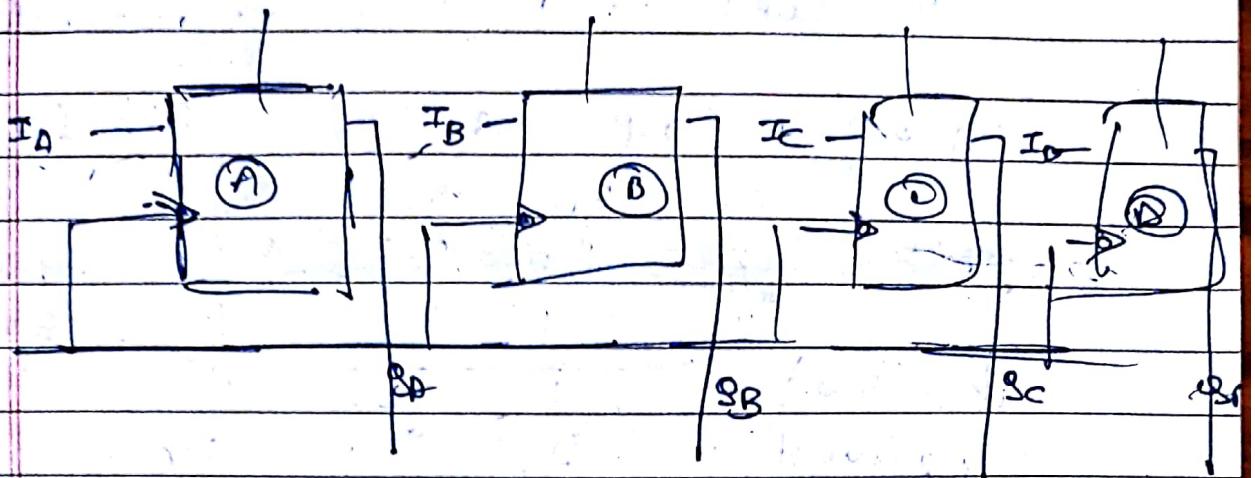
$\frac{1}{f}$ (Clock frequency)

Waveform for SISO / SIPO



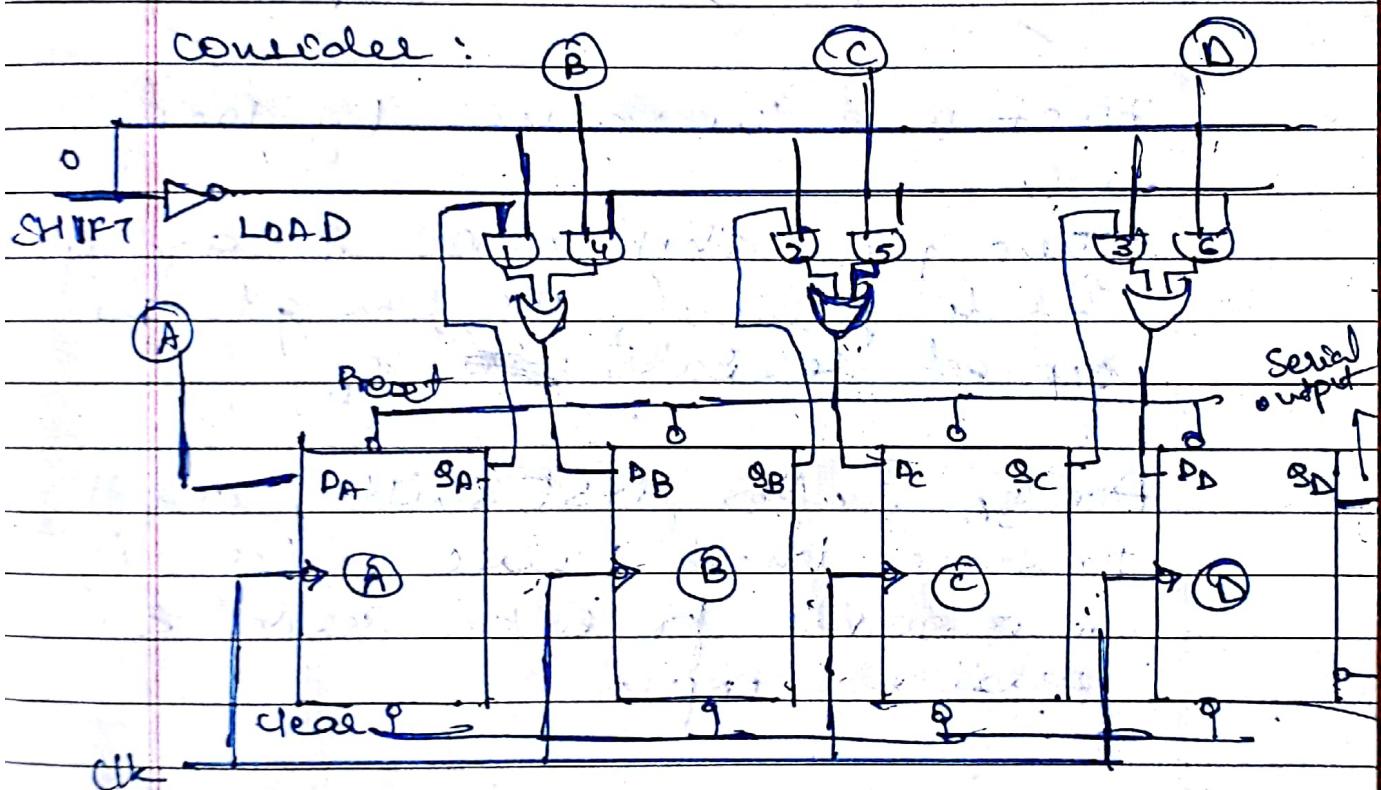
Parallel in parallel out:

Used when we want to give n bits of info + get n outputs at the same time, with some fixed delay for all, then we use PISO



PISO

Consider:



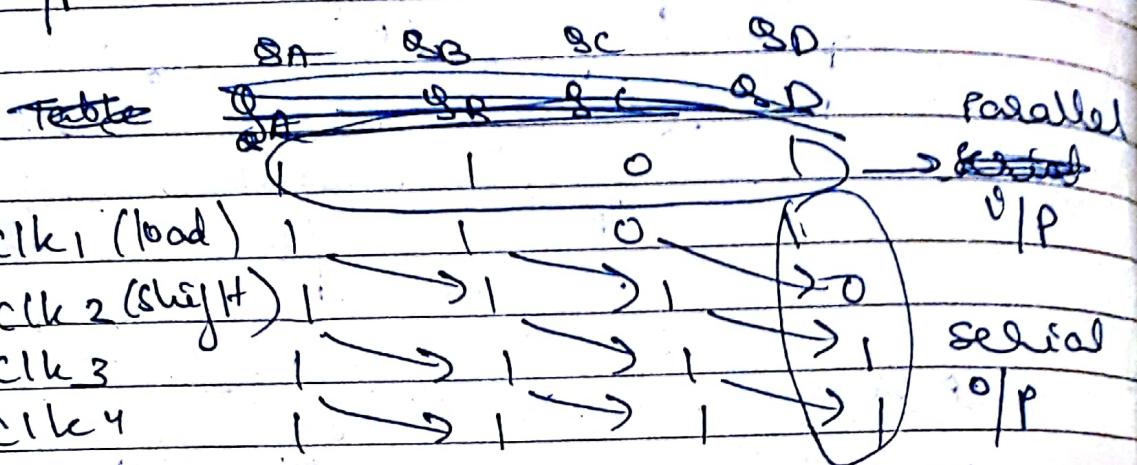
If SHIFT is activated (1)

1, 2, 3 → are activated

outputs of 1 → Q_A
 2 → Q_B
 3 → Q_C

Now, LOAD is 1, 4, 5, 6 activated.

O/p(8): 4 \rightarrow B, 5 \rightarrow C, 6 \rightarrow D.



First load the values by load mode.

Then, by shift mode & applying clock pulses, we can get the O/P ~~at~~ serially ~~at~~ Q_D.

For Q_A, after first shift, set it to some fixed value (here 1) (A is same for both serial & parallel input)

Universal shift Register:

Why 4x1 MUX

Since we are performing only 4 functions

(8 functions \Rightarrow can't use 8x1 MUX)

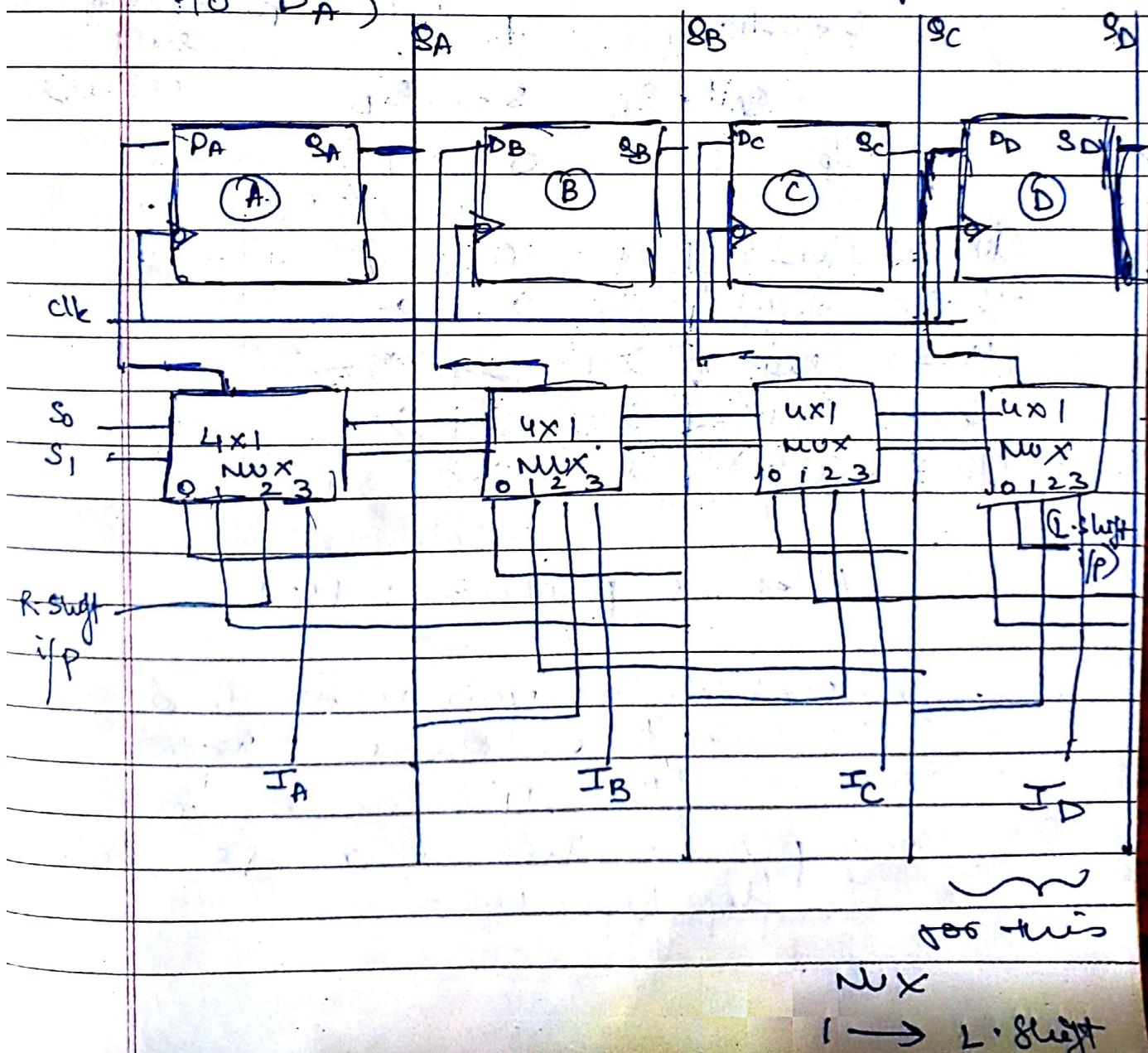
Functions :

Can be changed of also

| | |
|-----|-----------------------------|
| 0 0 | \rightarrow No change |
| 0 1 | \rightarrow Left shift |
| 1 0 | \rightarrow Right shift |
| 1 1 | \rightarrow Parallel load |

Parallel load : Giving i/p to the registers, giving clock pulse & that comes as output.

(Circular shift : Q_D output goes to P_A)

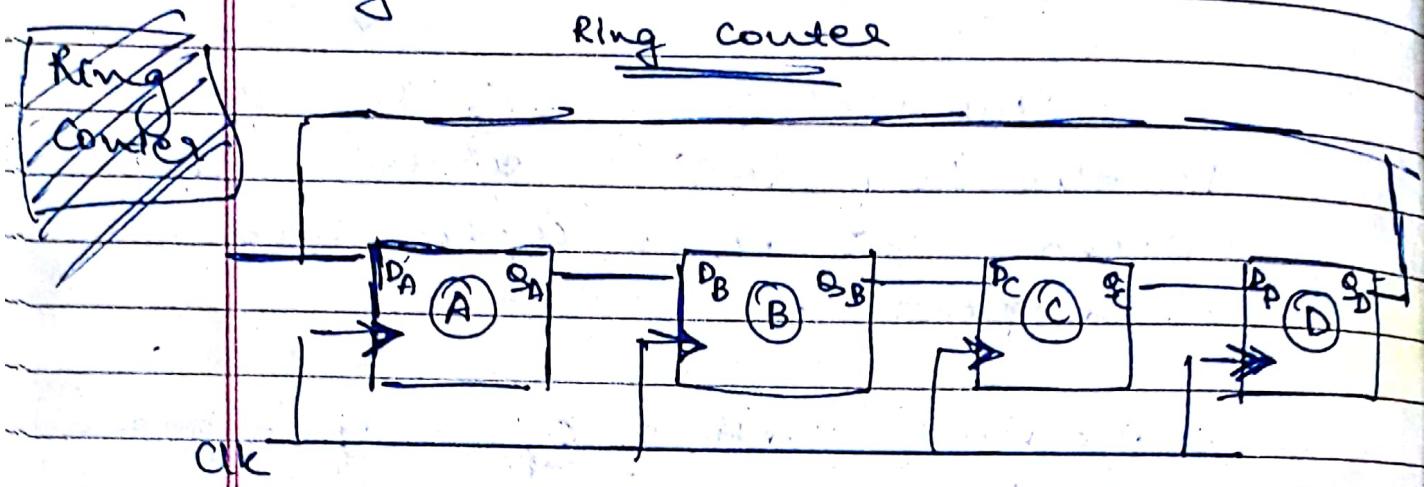


Scanned by CamScanner

It is called as universal shift register because it can be used for any application.

Ring & Johnson Counter:

Ring counter



Clk

Data : 1101

here circular
shift
occurs

Q_A Q_B Q_C Q_D

i/p : 1 1 0 1

Clk \rightarrow 1 (load) 1 1 0

Clk \rightarrow 2 (shift) 1 0 1 0

Clk \rightarrow 3 (shift) 0 1 1 0

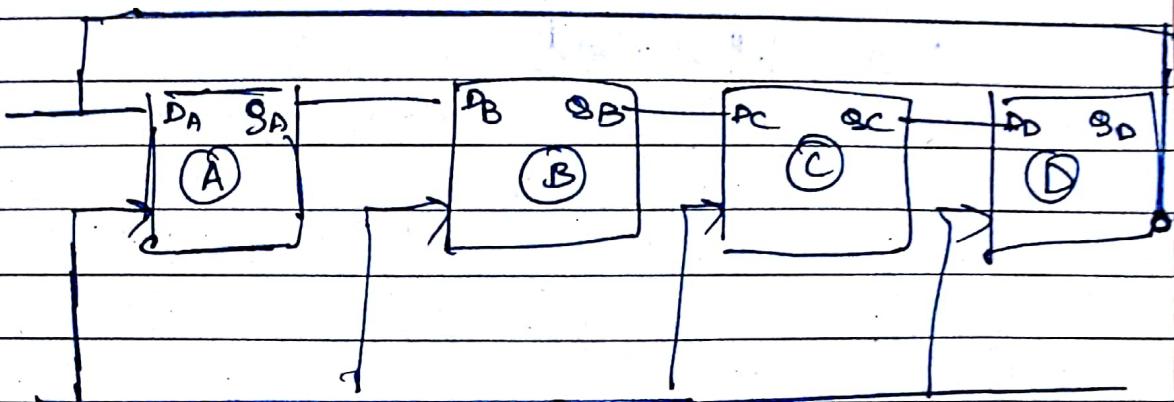
Clk \rightarrow 4 (shift) 0 0 1 1

Clk \rightarrow 5 (shift) 1 0 0 1

After 5 pulses : 1101

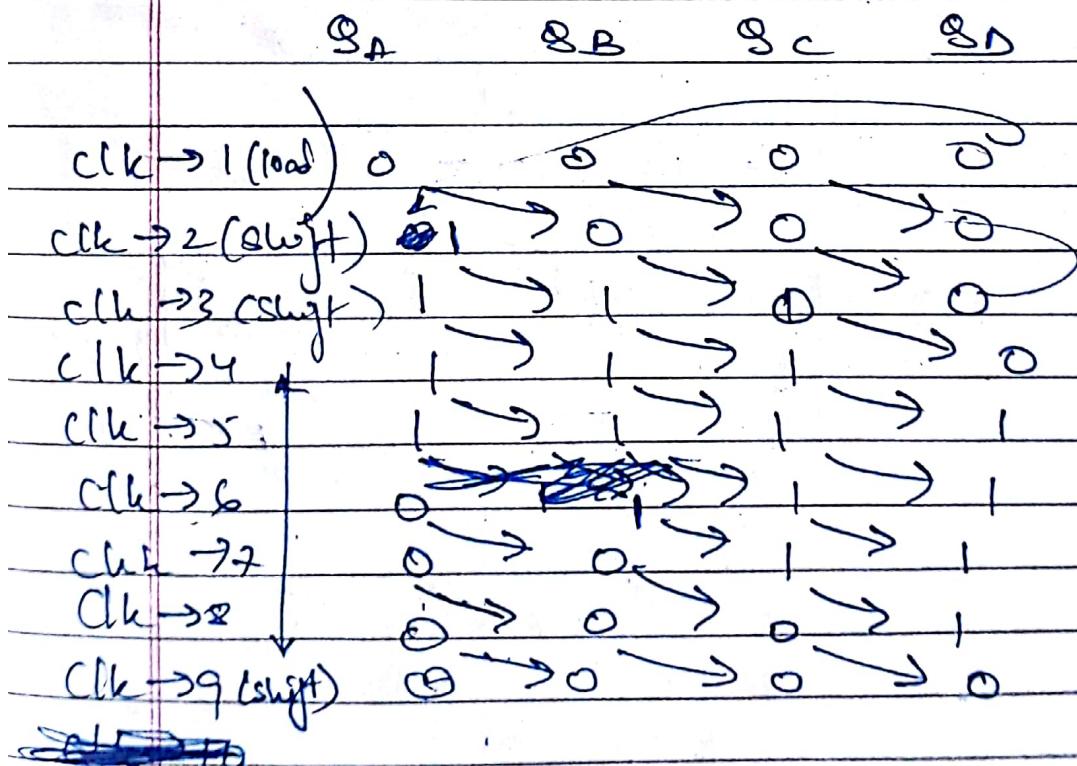
Whatever data you have loaded, to get back the same ~~input~~ data as o/p : you need n clock pulses where n is no. of flip flops used.

Johnson counter:



Data : 0000

~~Q_D~~ is connected to D_A



* Bidirectional shift register (using
the AND / OR gate network)
(both left & right)

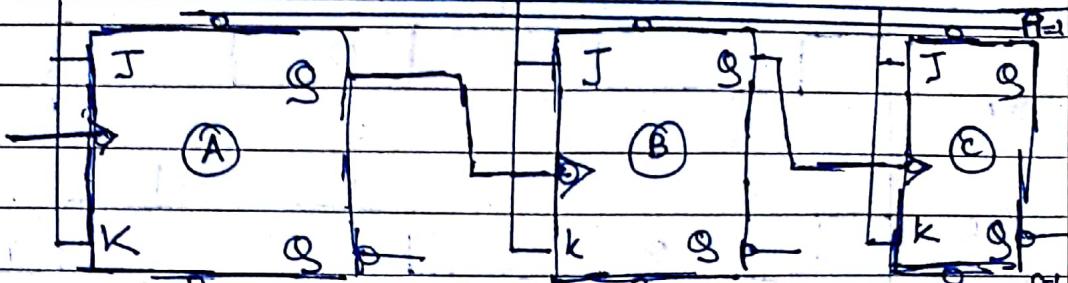
Unit 4

Counters

Counters → Asynchronous

→ Synchronous

Vcc



Design for asynchronous counter (fig(i))

Note: Counter: Device which is used for counting. It is of 2 types:

① Synchronous counters

(eg: Ring & Johnson counter)

② Asynchronous counters

(eg: ~~modular~~ up/down counter)

clock pulse

applies to

all flip flops

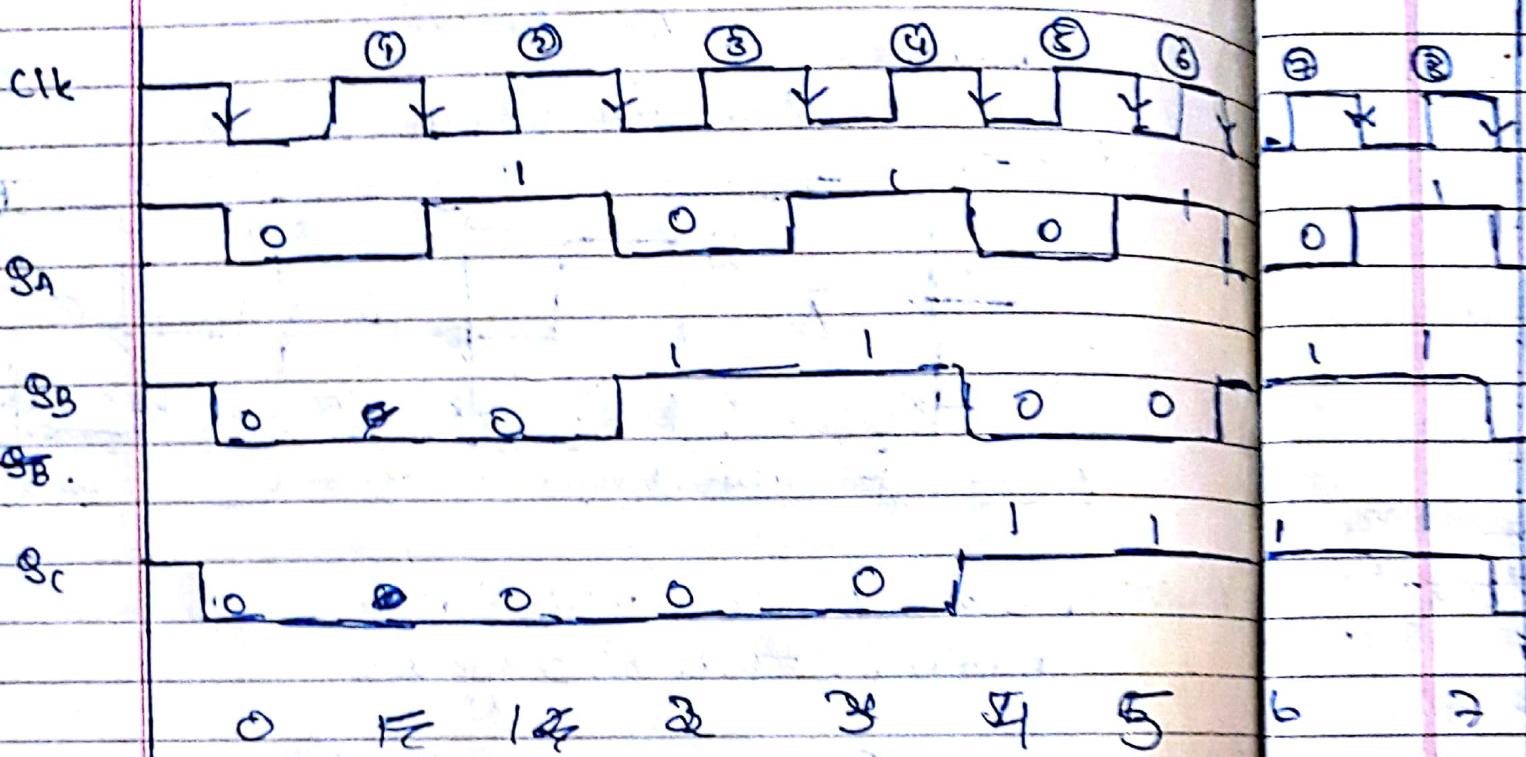
at the same

time.

↳ The o/p of first flip flop acts as clock to second flip flop etc.

The fig(i). shown uses -ve edge triggered flip flop. Both J & K are connected to Vcc → implies toggle state for the J-K flip flop at the -ve edge.

(Assume initial state is 1 for S_A ,
 $S_B, S_C \rightarrow$)
Waveshow for $S_A, S_B, S_C \leftarrow$



The asynchronous inputs are used
to count a specific no. of values
(ie; to break the counting)

flip flop ① : LSB

last flip flop : MSB

clock given to LSB flip flop

Nod or counters

Has n no. of counts.

eg: ~~111~~ nod = cont

(counts 5 values in general)

kg : To count from 0 to 4.

B A

0 0 0 }

0 0 1 }

0 1 0 }

0 1 1 }

1 0 0 }

1 0 1 }

1 1 0 }

1 1 1 }

Valid counts

Invalid counts

So consider the case
where we have to
go back

1 0 1



Take $Q_1 \cdot Q_2 = 0$

\therefore Till count = 4

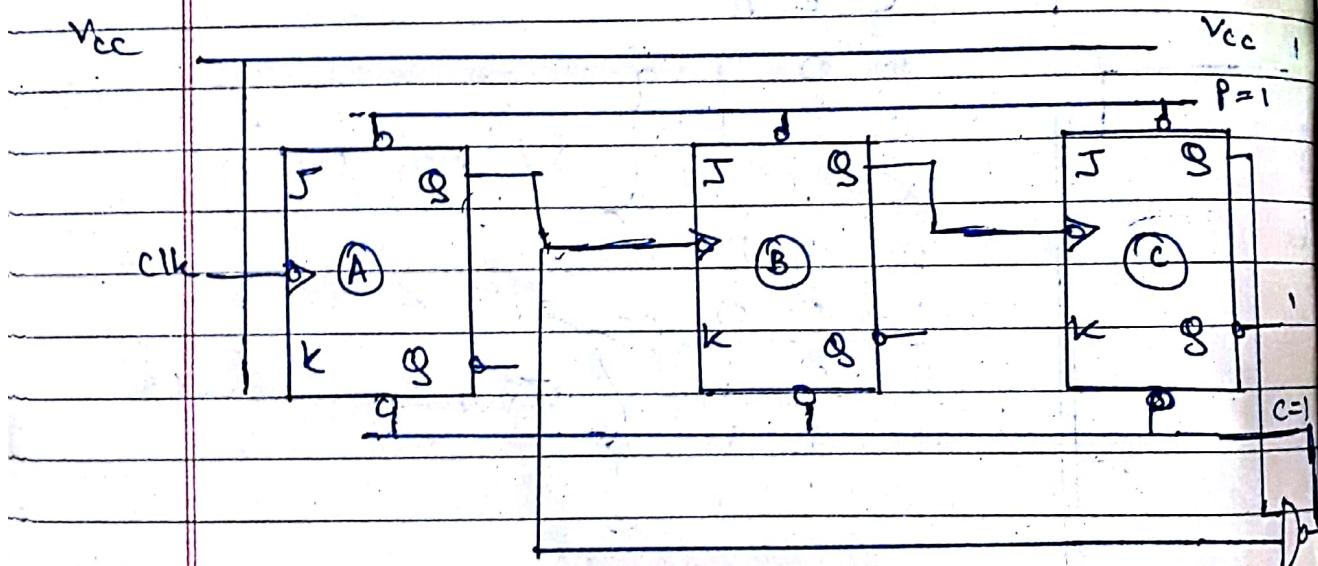
the clear value is
itself \Rightarrow functions like a
normal JK flip flop.

When count becomes 5:

immediately, $Q_a \cdot Q_c = 1$
clear. = 0

resets to 0.

Mod 5 ~~counter~~ asynchronous counter
(here up counter)



up counter: Counts from 0 to end value

Note: Asynchronous Counter can be designed JK & T flip flops mainly (Toggle counter)

even if with SR & D flip flop
can also be ~~counted~~ used; by
converting them to JK and or
T flip flop.

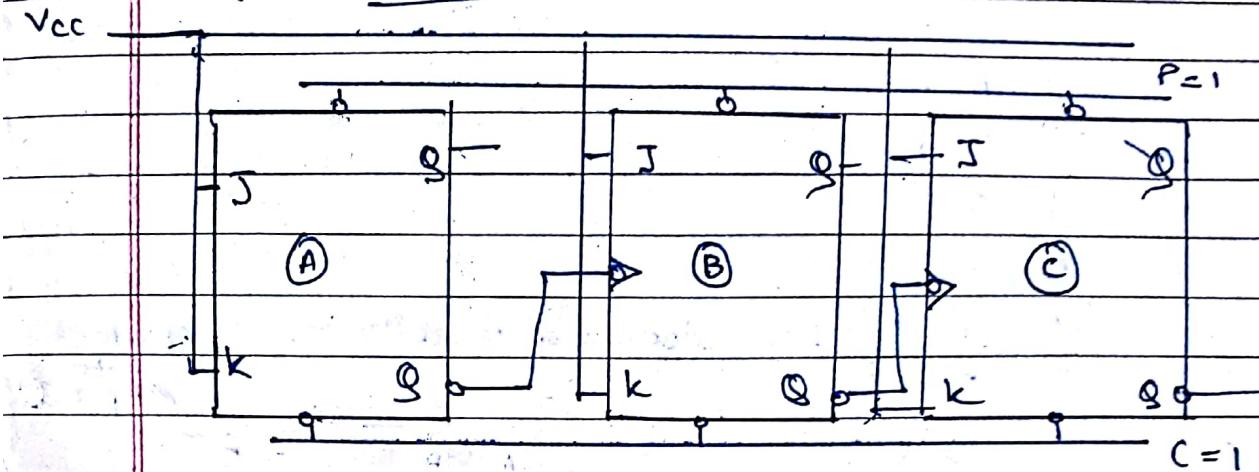
2 3 4



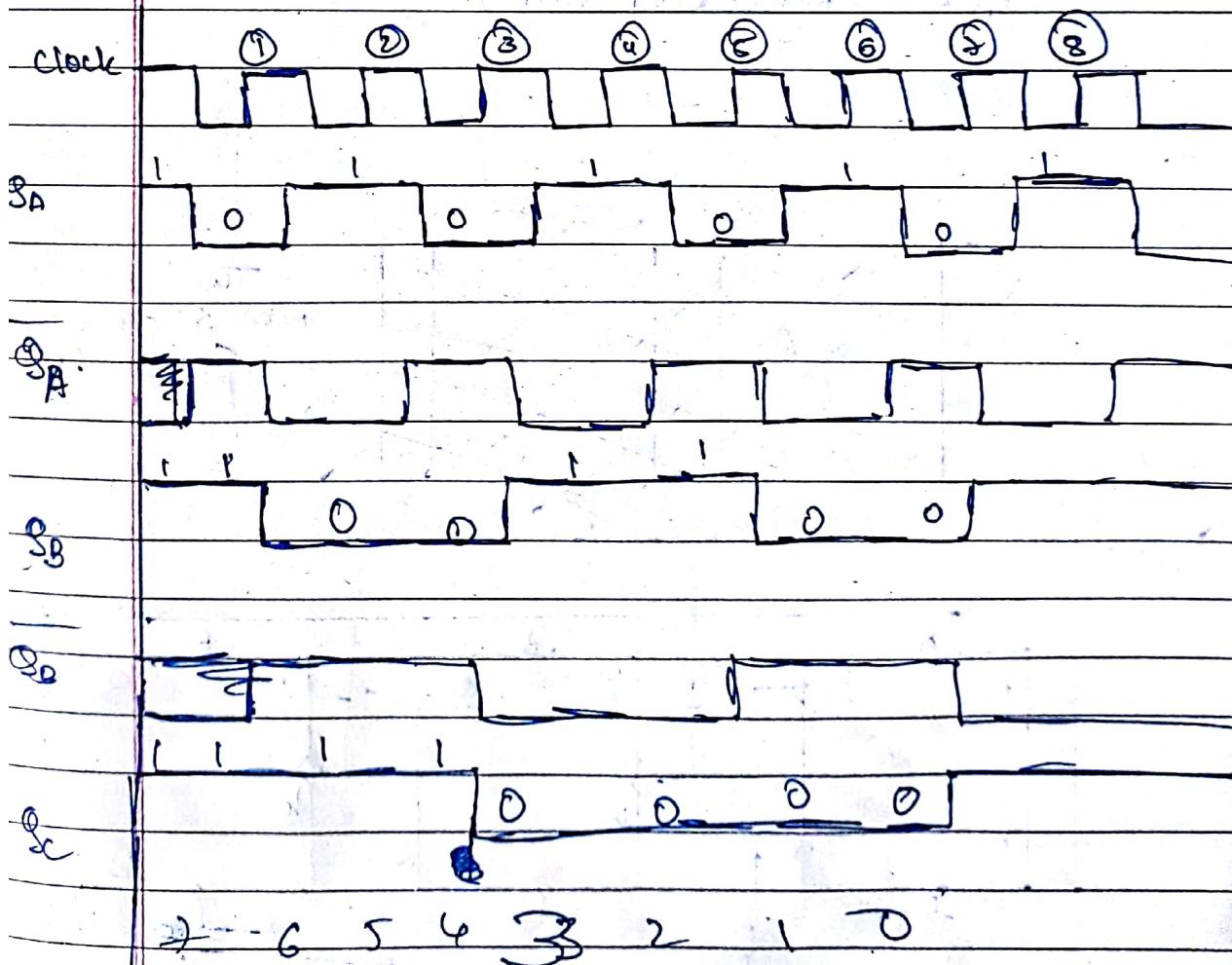
Date:

Page No.:

Down counter:



Wave diagram:



Counts from 7 to 0 7 to 0
↓ down ↑

Mod 5 down counter (asynchronous)

C B A

1 1 1

1 1 0

1 0 1

1 0 0

0 1 1

0 1 0 → Stopping condition

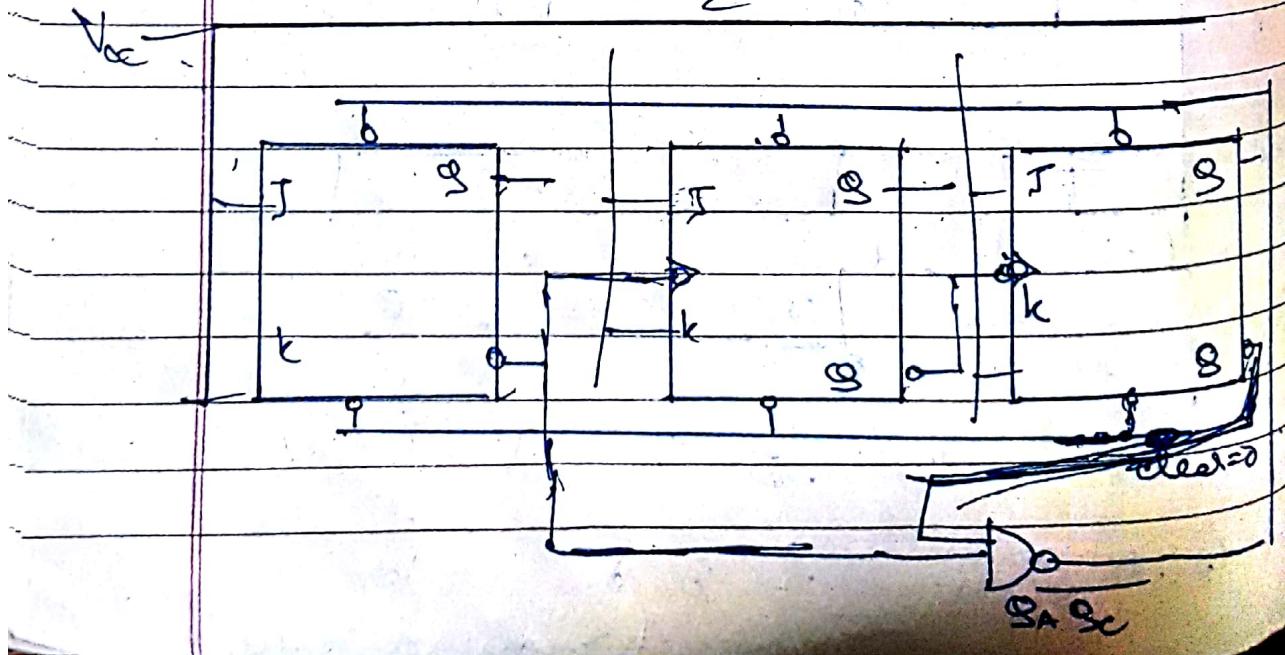
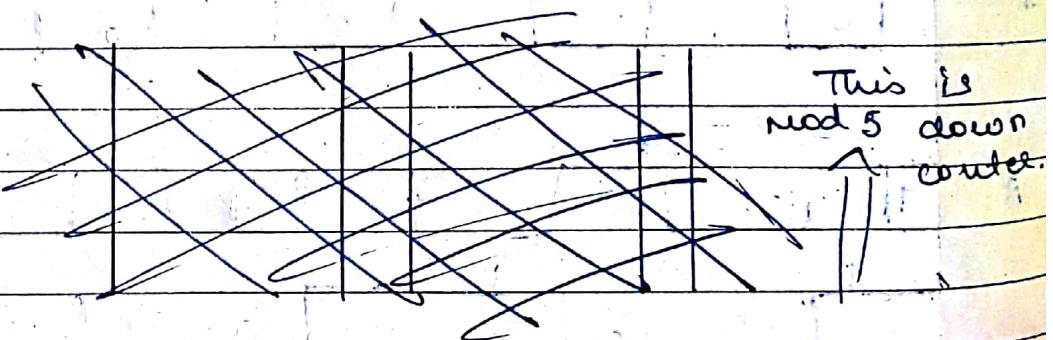
0 0 1

0 0 0

$\overline{Q}_A \overline{Q}_B \overline{Q}_C = 0$

connect
to
preset

To represent the function; prefer NAND gates
(other gates can also be used;
best preferably not)



Preset connects to gac

Synchronous counters:

(To ~~not~~ count change the counting sequence; ~~normal~~ not necessarily counting in order).

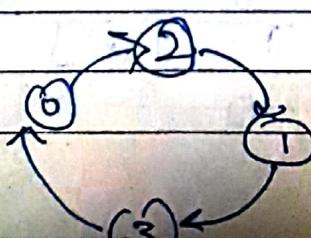
Eg For this we use excitation tables:

Jk flip flop

| SR | g | g^+ | S | R | g | g^+ | J | K |
|-----|---|-------|---|---|---|-------|---|---|
| 0 0 | 0 | 0 | 0 | x | 0 | 0 | 0 | x |
| 0 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | x |
| 1 0 | 1 | 0 | 0 | 1 | 1 | 0 | x | 1 |
| 1 1 | 1 | 1 | x | 0 | 1 | 1 | 0 | x |

| T & | g | g^+ | T | D |
|-----|---|-------|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |

Synchronous eg: Considering the sequence:



Here, we can use any flip flops:

Here, info is 2 bit
 $(0-3) \rightarrow 2$ flip flops enough.

| S_A | S_B | Q_A | Q_B | S_A | R_A | S_B | R_B |
|-------|-------|---------------------|-------|-------|-------|-------|-------|
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | X |
| | | $(0 \rightarrow 2)$ | | | | | |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| | | $(2 \rightarrow 1)$ | | | | | |
| 0 | 1 | 1 | 1 | 1 | 0 | X | 0 |
| | | $(1 \rightarrow 3)$ | | | | | |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| | | $(3 \rightarrow 0)$ | | | | | |

Now,

$$S_A = \overbrace{Q_B}^0 \quad \overbrace{Q_A}^1 = \overline{Q_A}$$

| | | |
|---|---|---|
| 0 | 1 | 1 |
| 0 | 0 | 0 |

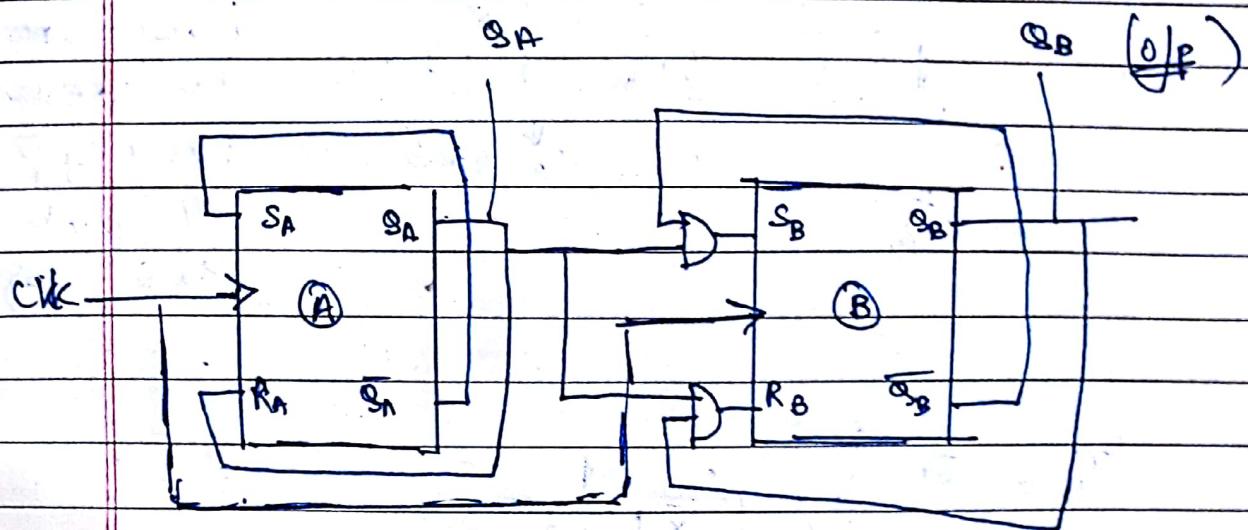
$$R_A = \overbrace{Q_B}^0 \quad \overbrace{Q_A}^1 \leftarrow Q_A$$

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |

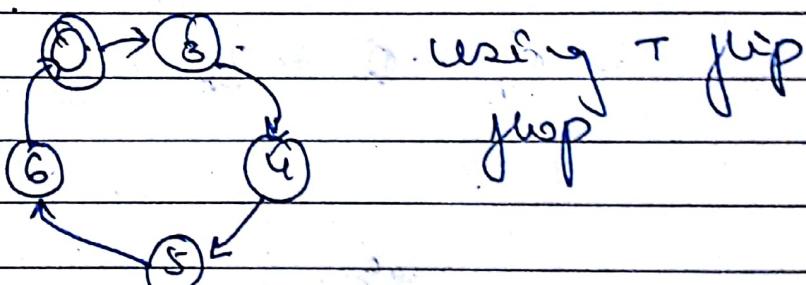
$$S_B = \overbrace{Q_A}^0 \quad \overbrace{Q_B}^1 \quad \left[\begin{array}{c|cc} 0 & 0 & X \\ 1 & 0 & 0 \end{array} \right] = Q_A Q_B$$

$$R_B = \overbrace{Q_A}^0 \quad \overbrace{Q_B}^1 \quad \left[\begin{array}{c|cc} 0 & 0 & 1 \\ 1 & X & 0 \end{array} \right] = Q_A Q_B$$

∴ This can be designed as:



Q. Design:



Ans.

| Q | Q^+ | T |
|-----|-------|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

(3 flip flops needed)

| Q_A | Q_B | Q_C | Q_A^+ | Q_B^+ | Q_C^+ | T_A | T_B | T_C |
|-------|-------|-------|---------|---------|---------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

$$T_A = \overline{Q_A} \begin{array}{c} Q_B Q_C \\ \hline 00 & 01 & 11 & 10 \end{array}$$

| | | | | |
|---|---|---|---|---|
| 0 | 0 | x | 1 | x |
| 1 | 0 | 1 | x | 1 |

$\downarrow Q_B \text{ (quad)}$

(for the
 $Q_A Q_B Q_C$
cases not
in sequence
(here 1, 2, 3)
 \hookrightarrow put
don't care)

$$= \overline{Q_B}$$

$$T_B = \overline{Q_A} \begin{array}{c} Q_B Q_C \\ \hline 00 & 01 & 11 & 10 \end{array}$$

| | | | | |
|---|---|---|---|-----|
| 0 | (| x | 1 |) x |
| 1 | 0 | 1 | x | 1 |

$$= \overline{\overline{Q_A}} + Q_C + Q_B$$

$$T_C = \overline{Q_A} \begin{array}{c} Q_B Q_C \\ \hline 00 & 01 & 11 & 10 \end{array}$$

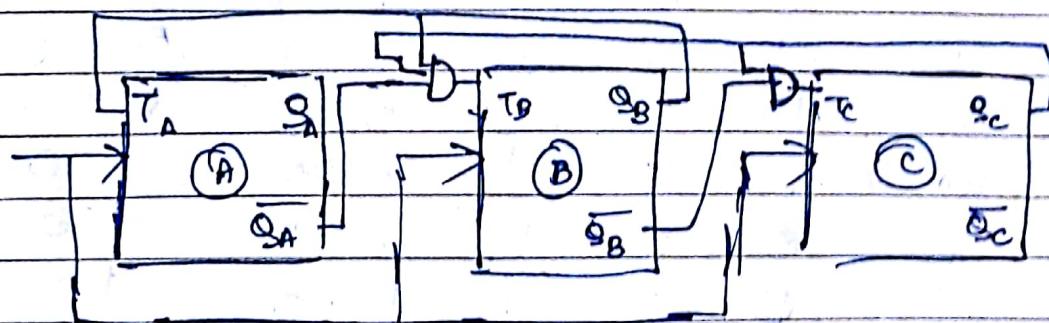
| | | | | |
|---|---|---|---|-----|
| 0 | (| x | 1 |) x |
| 1 | 1 | 1 | x | 0 |

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

$$\therefore T_A = \overline{\overline{Q_B}}$$

$$T_B = \overline{\overline{Q_A}} + Q_B + Q_C \quad T_C = \overline{\overline{Q_B}} + Q_C$$

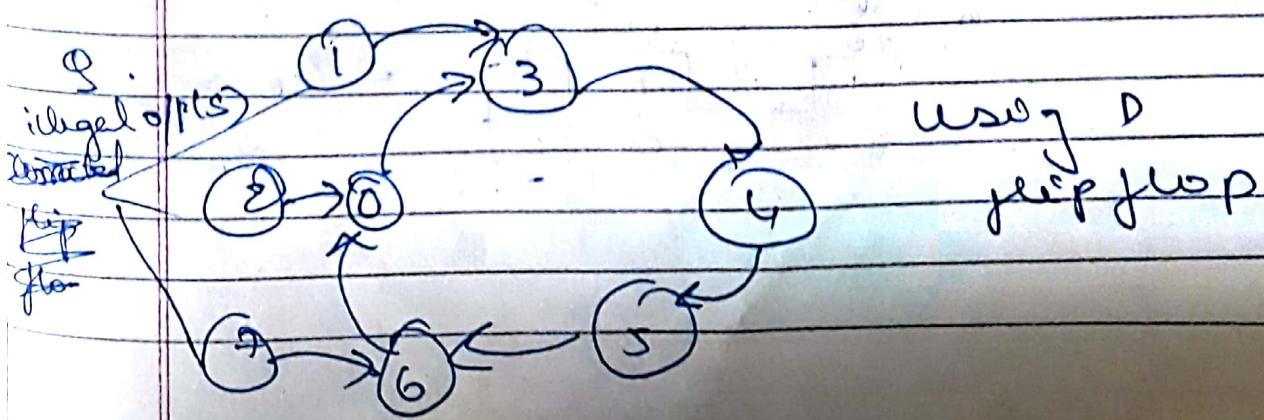
Design :



Self Connecting Counter:

Now, in this sequence, due to fluctuation, an illegal op may be obtained (for eg - for this case, 1 → 2 op) & may be obtained by mistake
 \hookrightarrow this will cause the sequence to halt.

To correct this, we can use a self-connecting counter, wherein the illegal op states will be mapped to some state in the sequence. (these connections can be given, if not given; then assume in any way).



| Q_A | Q_B | Q_C | $Q_A + Q_B$ | $Q_B + Q_C$ | D_A | D_B | D_C |
|-------|-------|-------|-------------|-------------|-------|-------|-------|
| 0 | 0 | 0 | 0 | + | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

(Note: This table can be written in order of the segments, or in ~~ascend~~ descending order with corresponding complement $Q_A^L Q_B^L Q_C^L$).

$$D_A = \overline{Q_A} \overline{Q_B} \overline{Q_C}$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |

$$= Q_A \overline{Q_B} + Q_B \overline{Q_C}$$

$$D_B = \overline{Q_A} \overline{Q_B}$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |

$$+ Q_A \overline{Q_C}$$

$$D_C = \overline{Q_B} \overline{Q_C}$$

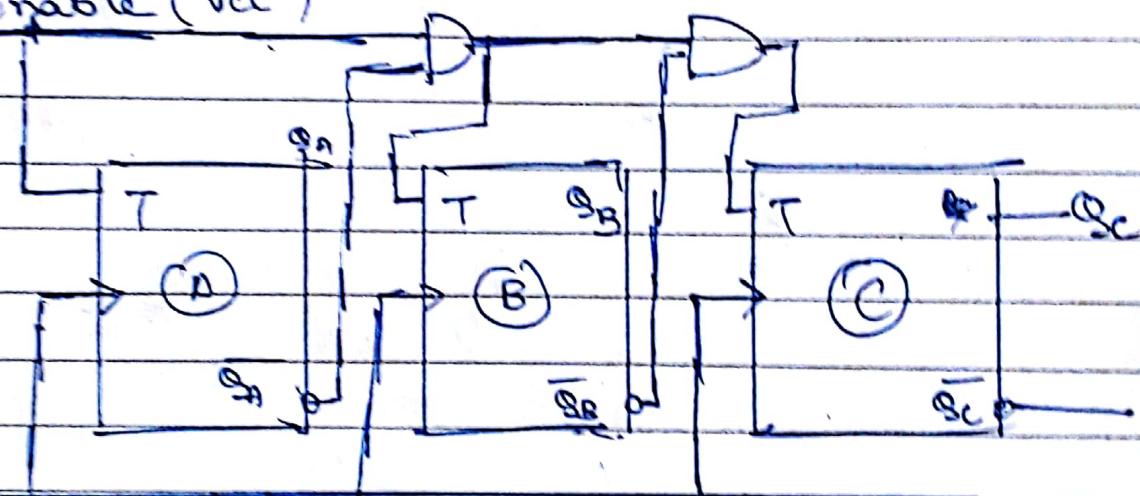
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |

$$= \overline{Q_B} \overline{Q_C}$$

$$+ \overline{Q_A} \overline{Q_B}$$

Design :

Q. Enable (Vcc)



(Earlier design of synchronous
counter)

FJ
a \bar{Q} \dot{Q}^+ qt

Q. Design 5421 code sequence using
JK flip flop.
~~(0 to 9 possibilities)~~

$$Q = 0 \quad \bar{Q} = 1 \quad S + RQ = \bar{Q}$$

$$S=0 \quad R=0$$

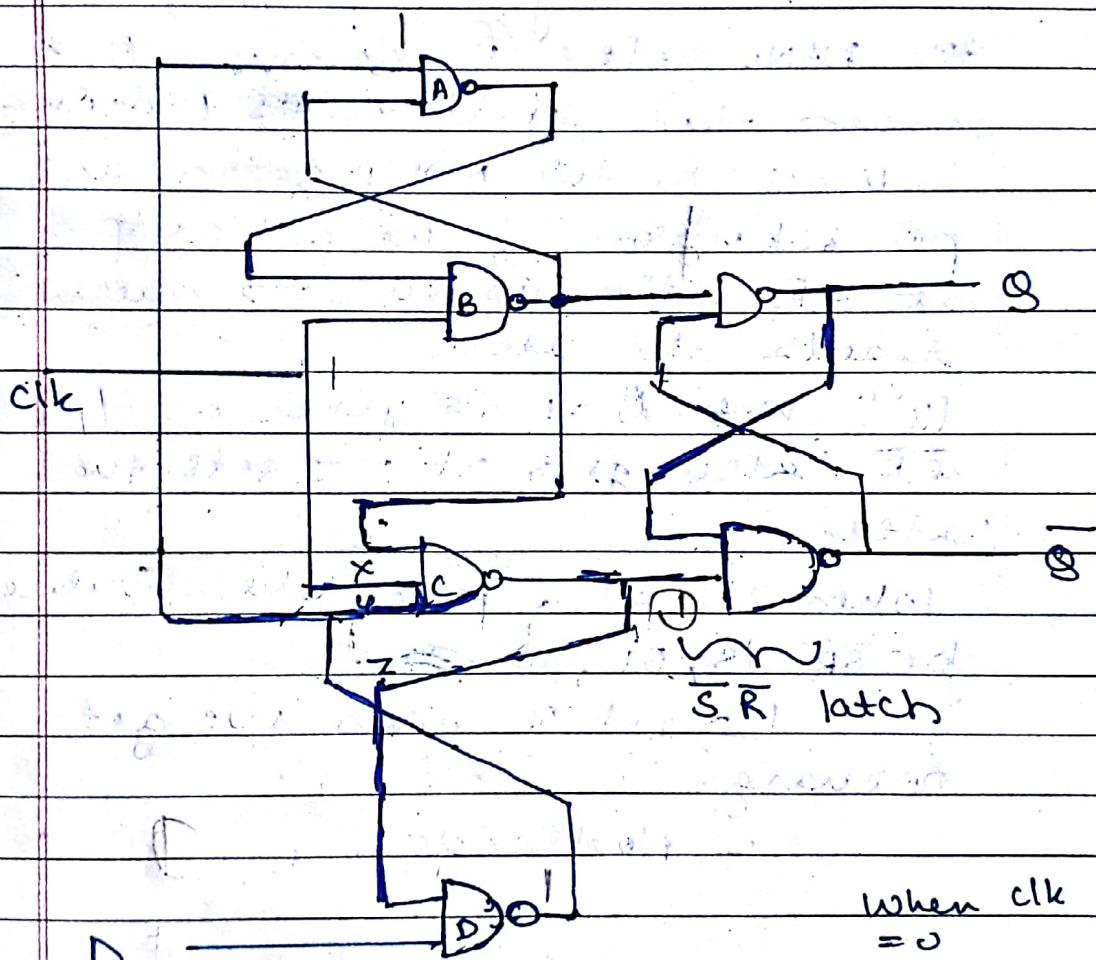
$$Q=1$$

$$Q+1 = 0$$

Date: _____
Page No.: _____

Unit 3 (cont.)

Positive edge triggered D flip flop



When $clk = 0$

B & C NAND gates directly give it

\Rightarrow No change

| D | clk | Q | Q^+ |
|---|-----------------------------|-----|-------|
| X | 0 | NC | NC |
| 0 | $\uparrow(0 \rightarrow 1)$ | 0 | 0 |
| 1 | $\uparrow(0 \rightarrow 1)$ | 1 | 1 |
| X | 1 | 0 | NC |

When the clk is changed from 0 \rightarrow 1 (positive edge on the timing diagram); until the other values are obtained, SR will have only

Initially, when $\text{clk} = 0$, $\overline{\text{SR}}$ latch gets 11
⇒ no change

when clk goes from 0 → 1 (+ve edge);
that time, the prev $\overline{\text{S}}$ & $\overline{\text{R}}$ values are
taken into consideration for the
circuit.

when $D = 0$ is given as i/p; the
NAND gate ① becomes 0 &
passes this about - ~~the~~ Pulsing
 $\text{clock} = 1$ for the NAND gates as
per selection; we get output
at $\overline{\text{SR}}$ latch as 10 → which
resets the latch.

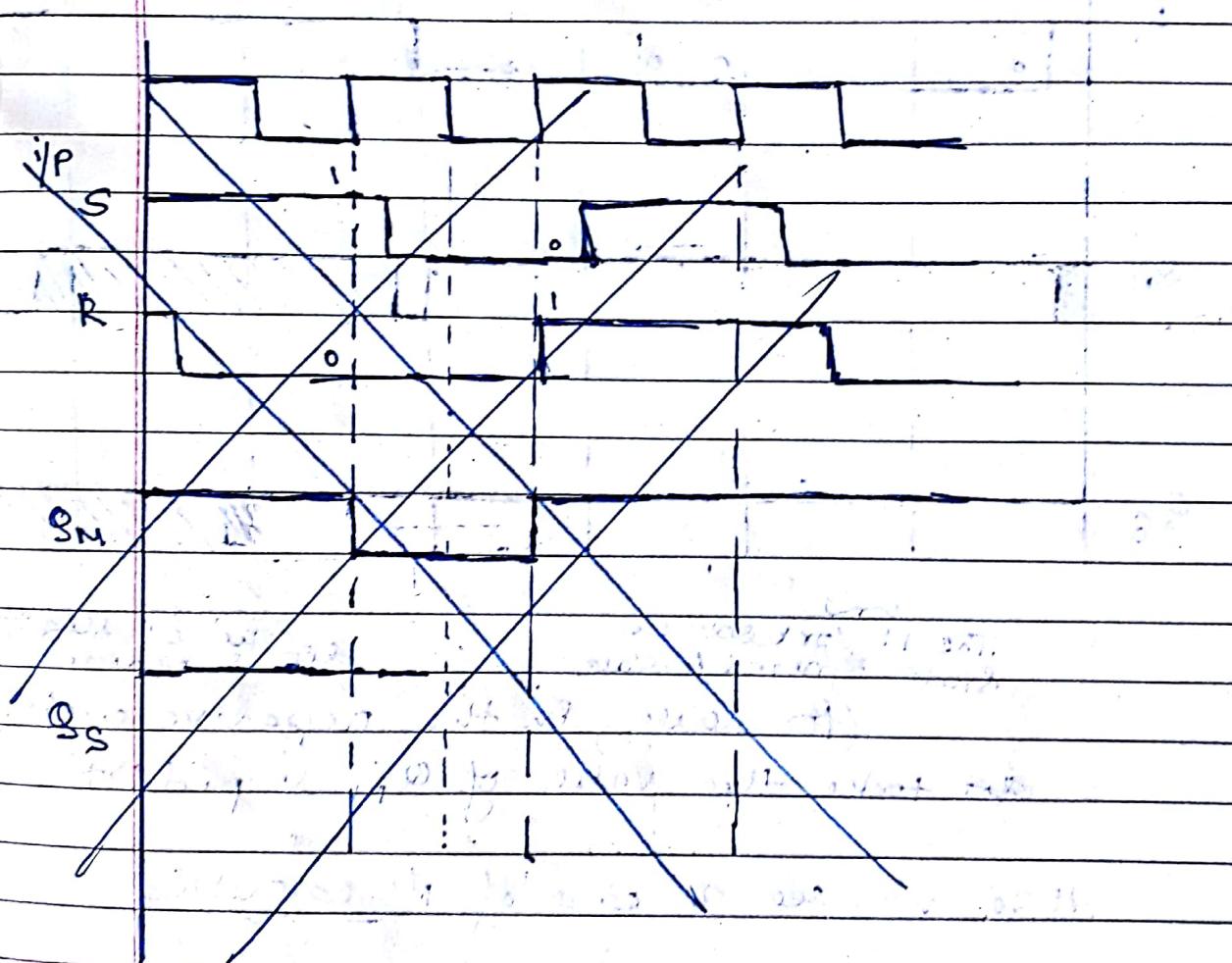
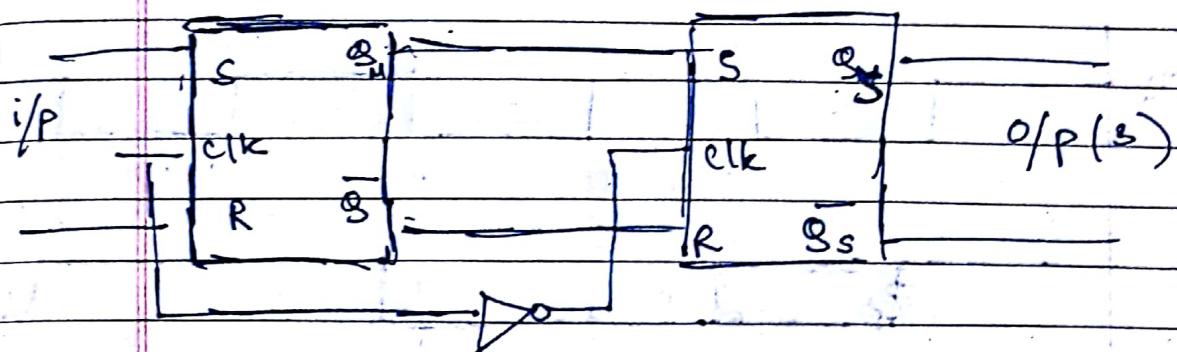
Similarly, when $D = 1$ is given as i/p,
 $\overline{\text{SR}}$ latch gets 01 → sets the
latch.

when $\text{clk} = 1$, possible value
for $\overline{\text{SR}}$ is 01, 10 ~~11~~.

→ all 3 cases we get
no change.

∴ No change.

SR Master slave

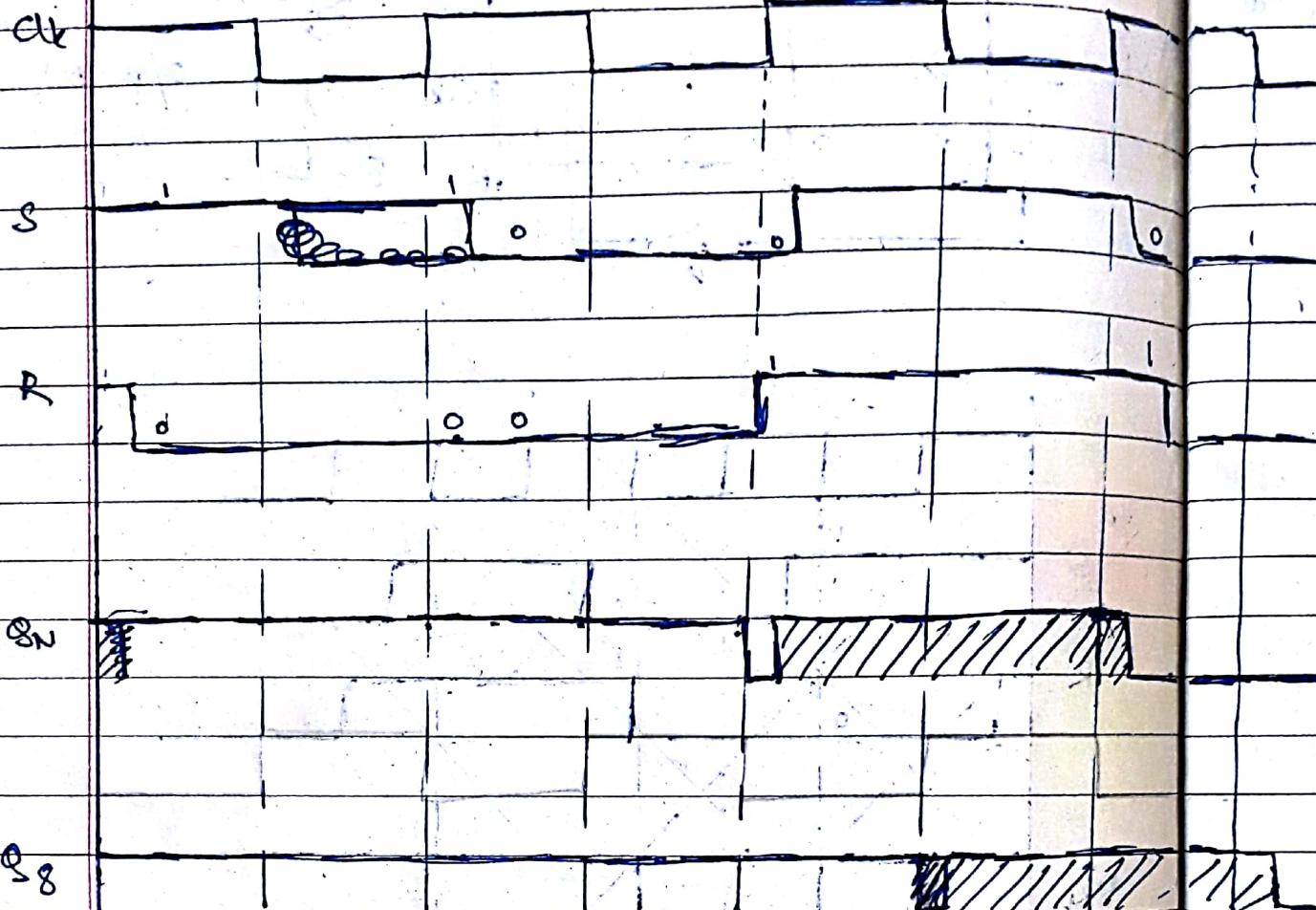


\Rightarrow initial assumption

Q_S

PTO 1

0.0 forbidden state : X X X X X



The 11 forbidden state ignored in slave
(for slave, in the negative cycle,
do take the value of Q_N & put it)

The 01 slight dip is ignored

Here, we see a case of 1's catching

How to avoid 0's and 1's catching
(JK master slave)

~~No JK~~ In order to avoid,
the J & K input values are
kept constant when the clock

①

②

$$Q \bar{Q} = 1$$

$$Q \bar{Q} = 0$$

SR flip flop

00

01

10

11

No Date:

Reset P No:

Set

FS

is at logic 1 (i.e., when master is enabled), ~~at~~

{Alternatively, this problem can be overcome if the master responds during the +ve or rising edge of the clock & the state of the master be transferred to the slave at the -ve or falling edge of the clock.}

↳ Data lockout

(edge triggered master slave)

Keynote:

Summary for Test 2:

①

PLD's \rightarrow PROM, PLA, PAL.

~~also~~.

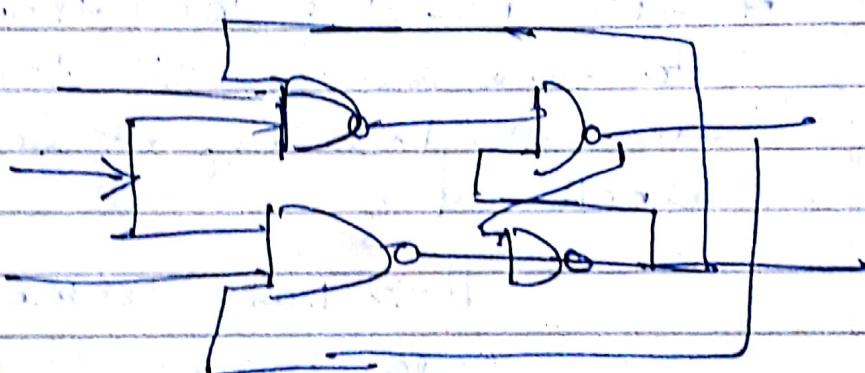
Catalogue go for simplified/minimized expr.)

②

Bistable element ~~to~~, Latch (SR, \bar{SR}), flip flops, applications of SR latch (switch debouncing)

(B) construction: SR Master slave, JK Master slave, D flip flop (edge triggered)

If they ask to draw 1ve edge triggered JK or SR, just draw the pulse triggered flip flop, write the truth table for edge triggered & draw the circuit & charge clock in circuit



1ve edge triggered JK

If they ask any flip flop: write 1 advantage also,

(Note disadv. of M slave!)

① 0's & 1's catching

② Delayed o/p.

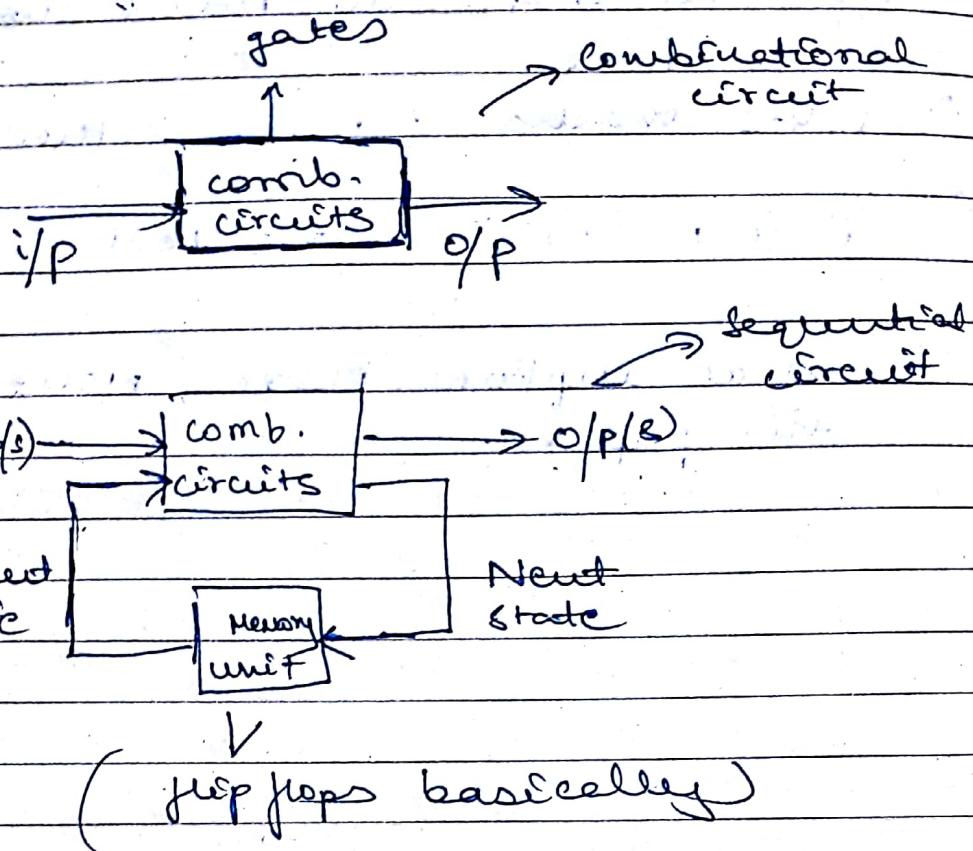
Real time applications can also be asked.

If they ask say need 5 asynchronous counters.

- (i) if 4-5 weeks : write either up/down
- (ii) if 10 weeks : write both.

Any assumptions made ; mention on the paper.

Unit 5:



Designing ~~sequential~~ synchronous circuits.

Types

- I. Design a synchronous sequential circuit using +ve edge triggered JK flip flop with minimal combinational gating to generate the following! (use reduced expression)

1. $0 - 1 - 2 - 0$ if $I/P \times = 0$ } pattern part
 is 1110 to
 0 - 2 - 1 - 0 if $I/P \times = 1$ } counter part

Provide an output which goes

high to indicate the non-zero states in $x=0$ sequence

Ans. In the pattern : maximum
no = 2

∴ 2 bits necessary
→ 2 flip flops.

Table :

| S | S^+ | J | K |
|---|-------|---|---|
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | 1 | X |
| 1 | 1 | 0 | X |

| X | A | B | A^+ | B^+ | J_A | K_A | J_B | K_B | o/p |
|---|---|---|-------|-------|-------|-------|-------|-------|-----|
| 0 | 0 | 0 | 0 | 1 | 0 | X | 1 | X | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | X | X | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | X | 1 | 0 | X | 1 |
| 0 | 1 | 1 | — | — | X | X | X | X | X |
| 1 | 0 | 0 | 1 | 0 | X | 0 | X | 0 | 0 |
| 1 | 0 | 1 | 0 | X | 0 | X | X | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | X | 1 | 1 | X | 0 |
| 1 | 1 | 1 | — | — | X | X | X | X | X |
| Don't care condition (not valid) Invalid cases | | | | | | | | | |

o/p depends on present states.

- * for a valid sequence, if no o/p is defined given → assume ①
- * for an invalid sequence, o/p is X.

Equations:

J_A :

| x | $\bar{A}B$ | 00 | 01 | 11 | 10 |
|-----|------------|----|---------|----|----|
| 0 | - | 0 | (1 X) | X | |
| 1 | - | 1 | 0 X | X | |

$$J_A = \bar{B}\bar{x} + \bar{B}x = \underline{\underline{B \oplus x}}$$

K_A :

| x | $\bar{A}B$ | 00 | 01 | 11 | 10 |
|-----|------------|---------------|----|----|----|
| 0 | - | X X X 1 | | | |
| 1 | - | X X X 1 | | | |

$$K_A = \underline{\underline{1}}$$

J_B :

| x | $\bar{A}B$ | 00 | 01 | 11 | 10 |
|-----|------------|-----------------|----|----|----|
| 0 | - | (1 X) X 0 | | | |
| 1 | - | 0 X X 1 | | | |

$$J_B = \bar{A}\bar{x} + Ax = \underline{\underline{A \odot x}}$$

K_B :

| x | $\bar{A}B$ | 00 | 01 | 11 | 10 |
|-----|------------|---------------|----|----|----|
| 0 | - | X 1 X X | | | |
| 1 | - | X 1 X X | | | |

$$K_B = \underline{\underline{1}}$$

O/P:

| x | $\bar{A}B$ | 00 | 01 | 11 | 10 |
|-----|------------|-----------------|----|----|----|
| 0 | - | 0 (1 X) T | | | |
| 1 | - | 0 0 X 0 | | | |

$$\text{O/P} = \bar{x}B + \bar{x}$$

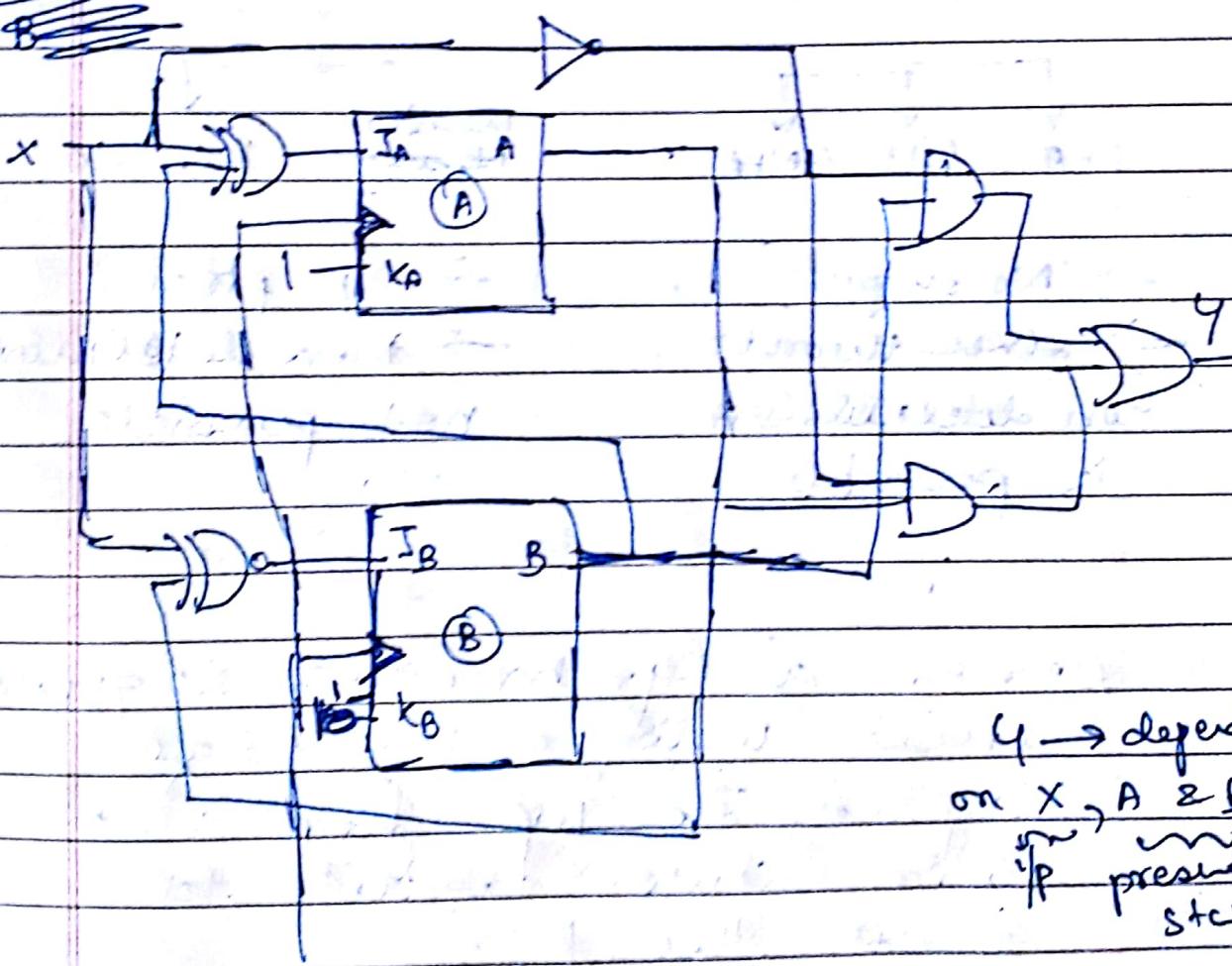
$$I_A = B \oplus X \quad K_A = 1$$

$$I_B = A \odot X \quad K_B = 1$$

$$O/P = \overline{X}B + \overline{X}A.$$

Circuit:

~~X~~
~~A~~
~~B~~



$Y \rightarrow$ depends
on $X, A \& B$
if present
state

Clock

Combinational
circuit
unit
(flip flop
part)

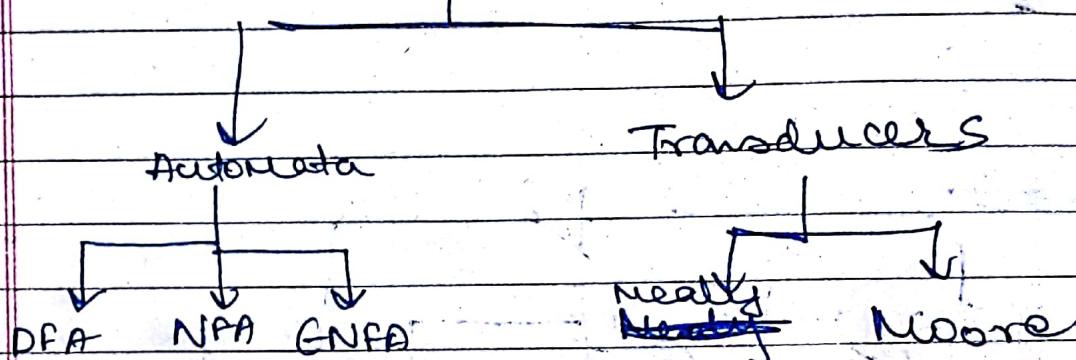
Combinational
circuit

This type of circuit :

Combinational cir + NUV + Comb cir
+ Y depends on present state
 \rightarrow Mealy Machine

Note:

Finite Machines



- \rightarrow No output
- \Rightarrow Since theoretic, non determinism is possible
- \rightarrow O/p present
- \rightarrow Non determinism not possible.

Q Design a synchronous sequential circuit using a +ve edge triggered JK flip flop with min - comb. circuit to generate the foll:

0-1-2-0 $x=0$

0-21-0 $x=1$

The output now goes to high whenever the circuit is in non-zero states irrespective of the sequence

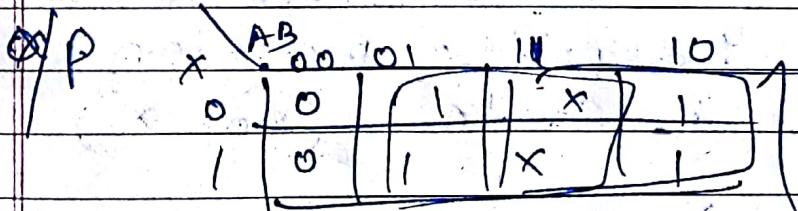
Ans.

| x | A | B | A^+ | B^+ | J_A | K_A | J_B | K_B | O/P |
|-----|-----|-----|-------|-------|-------|-------|-------|-------|-----|
| 0 | 0 0 | 0 0 | 0 1 | 1 0 | 0 X | 1 X | 1 X | 0 | 0 |
| 0 | 0 1 | 1 0 | 1 0 | 0 1 | X | X | X | 1 | 1 |
| 0 | 1 0 | 0 0 | 0 0 | 0 0 | X 1 | 0 X | 0 X | 1 | 1 |
| 0 | 1 1 | 0 0 | 0 0 | 0 0 | X X | X X | X X | X | X |
| 1 | 0 0 | 1 0 | 1 0 | 0 1 | 1 X | 0 X | 0 X | 0 | 0 |
| 1 | 0 1 | 0 0 | 0 0 | 0 0 | 0 X | X 1 | X 1 | 1 | 1 |
| 1 | 1 0 | 0 1 | 0 1 | 1 0 | X 1 | 1 X | 1 X | 1 | 1 |
| 1 | 1 1 | — — | — — | — — | X X | X X | X X | X X | X X |

J_A , K_A , J_B & K_B same as last time

$$J_A = B \oplus X \quad K_A = 1$$

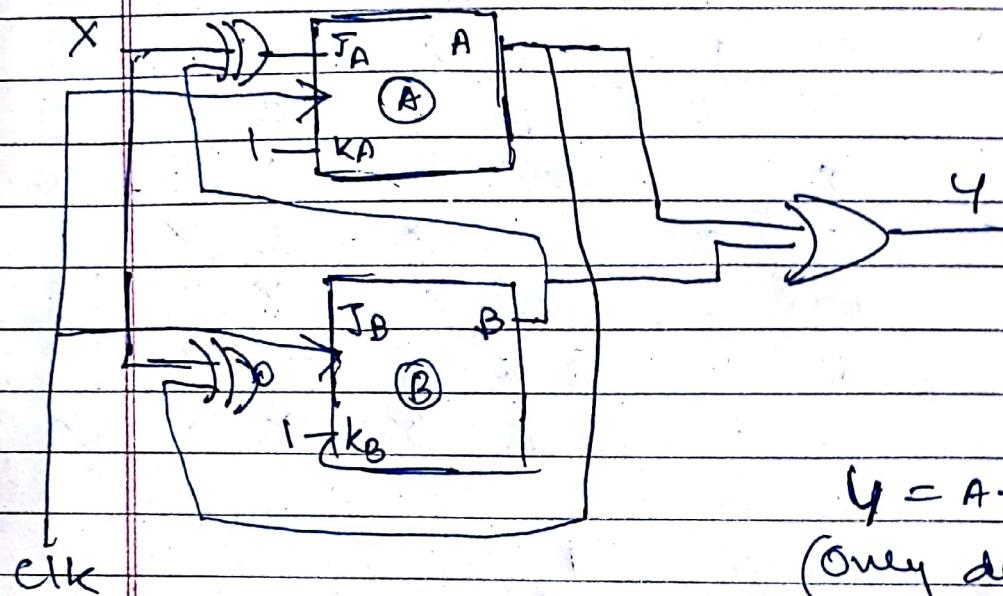
$$J_B = A \odot X \quad K_B = 1$$



$$O/P = \underline{A + B}$$

Pro

Circuit :



$$Y = A + B$$

(Only depends on
present
states)

clk
~~~~~  
Comb.  
circuit

~~~~~  
Memory
unit

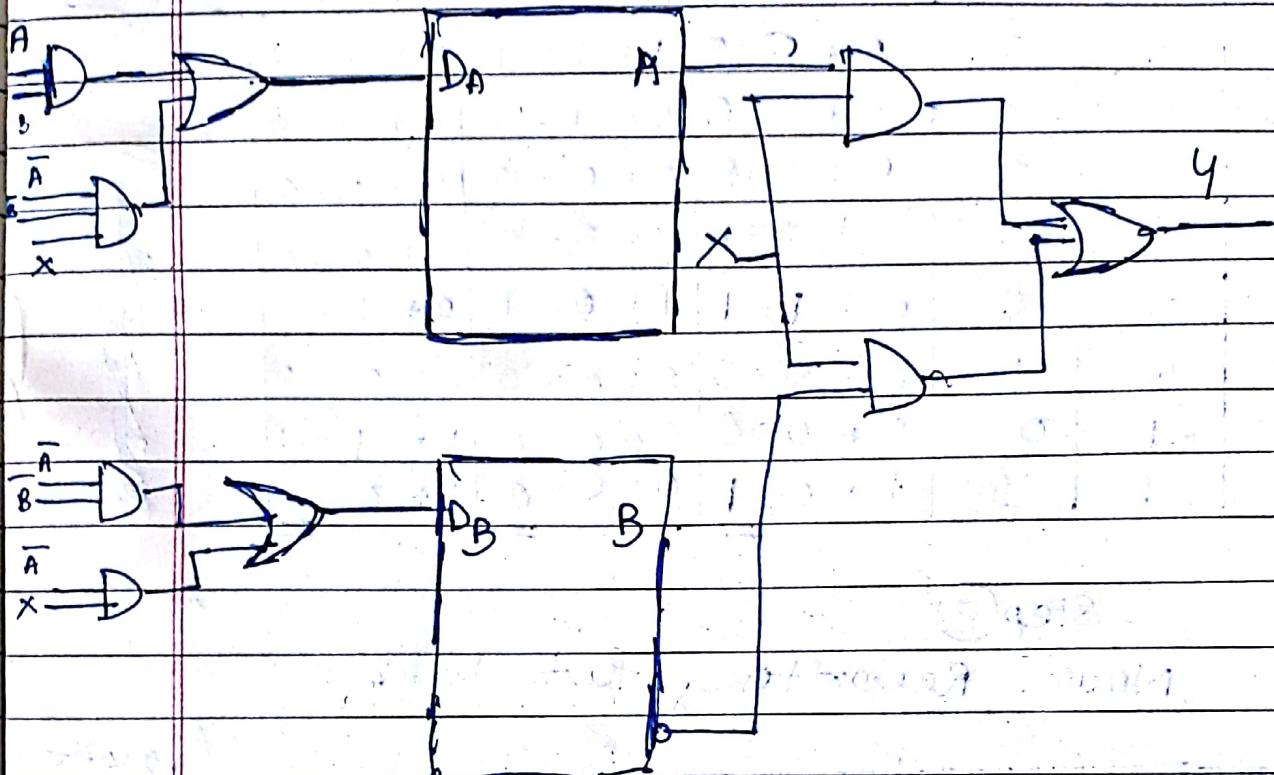
~~~~~  
Combination  
circuit

Comb. Circuit + Memory unit + Comb. circuit  
+  $Y$  depends only on  
present state (not  $X$ )

⇒ Noore Machine

writing State diagram  
(sequence)

Q. Consider the following:



Draw state diagram

Ans.

Here A & B are S/P(s) of flip flops.

from the circuit: Step ①

$$D_A = AB + \overline{A}\overline{B}X$$

$$D_B = \overline{AB} + \overline{A}X$$

$$Y = AX + \overline{B}X$$

Do substitution of the values to get  $D_A$ ,  $D_B$ ,  $y$

Step ②

| $X$ | A | B | $D_A$<br>$(AB + \bar{A}\bar{B}X)$ | $D_B$<br>$(\bar{A}\bar{B} + \bar{A}X)$ | $y$<br>$(AX + X\bar{B})$ |
|-----|---|---|-----------------------------------|----------------------------------------|--------------------------|
| 0   | 0 | 0 | $0 + 0 = 0$                       | $1 + 1 = 1$                            | $0 + 0 = 0$              |
| 0   | 0 | 1 | $0 + 0 = 0$                       | $0 + 1 = 1$                            | $0 + 0 = 0$              |
| 0   | 1 | 0 | $0 + 0 = 0$                       | $0 + 0 = 0$                            | $0 + 0 = 0$              |
| 0   | 1 | 1 | $1 + 0 = 1$                       | $0 + 0 = 0$                            | $0 + 0 = 0$              |
| 1   | 0 | 0 | $0 + 1 = 1$                       | $1 + 0 = 1$                            | $0 + 1 = 1$              |
| 1   | 0 | 1 | $0 + 0 = 0$                       | $0 + 0 = 0$                            | $0 + 0 = 0$              |
| 1   | 1 | 0 | $0 + 0 = 0$                       | $0 + 0 = 0$                            | $1 + 1 = 1$              |
| 1   | 1 | 1 | $1 + 0 = 1$                       | $0 + 0 = 0$                            | $1 + 0 = 1$              |

Step ③

Now, rewriting this tab:

| <del>PS</del> | $X=0$ | $X=1$ | $y$   |       |
|---------------|-------|-------|-------|-------|
| A             | $D_A$ | $D_B$ | $x=0$ | $x=1$ |
| 0             | 0     | 1     | 1     | 0     |
| 0             | 1     | 0     | 0     | 0     |
| 1             | 0     | 0     | 0     | 0     |
| 1             | 1     | 0     | 0     | 1     |

Transition  
Table

In a D flip flop,  $D_A$ ,  $D_B$  are

the inputs

&  $A^+$ ,  $B^+ = \cancel{D_A^+}, \cancel{D_B^+}$

itself for D flip

: final step : Step ④

| Present State  |                | Next State     |                | Y              |                | Excitation Table |
|----------------|----------------|----------------|----------------|----------------|----------------|------------------|
| A              | B              | $x=0$          | $x=1$          | $x=0$          | $x=1$          |                  |
| A <sup>+</sup> | B <sup>+</sup> | A <sup>+</sup> | B <sup>+</sup> | A <sup>+</sup> | B <sup>+</sup> |                  |
| 0              | 0              | 0              | 1              | 1              | 1              | 0                |
| 0              | 1              | 0              | 1              | 0              | 0              | 0                |
| 1              | 0              | 0              | 0              | 0              | 0              | 1                |
| 1              | 1              | 1              | 0              | 1              | 0              | 1                |

Step ⑤ Assume :

$$\begin{array}{l} 00 \rightarrow a \\ 01 \rightarrow b \\ 10 \rightarrow c \\ 11 \rightarrow d \end{array}$$

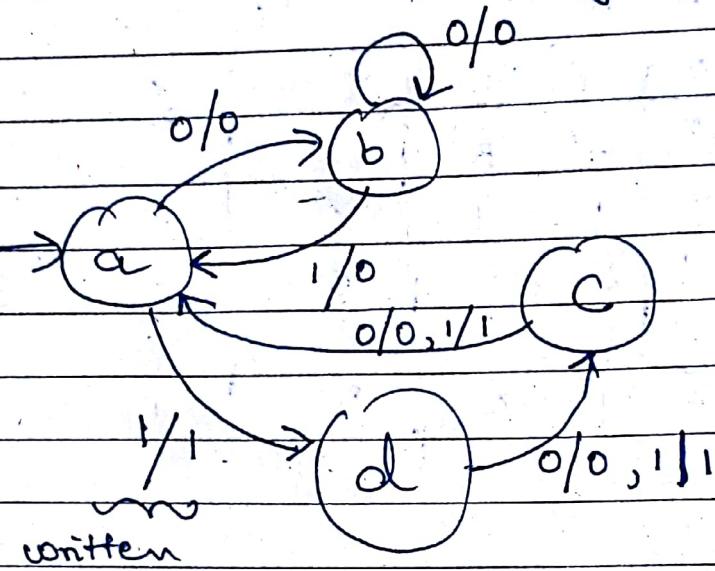
\* - We get : (State Table)

| PS                          | NS    | Y     |       |
|-----------------------------|-------|-------|-------|
| Consider<br>↑ ps<br>a start | $x=0$ | $x=1$ | $x=0$ |
| a                           | b     | d     | 0     |
| b                           | b     | a     | 0     |
| c                           | a     | a     | 0     |
| d                           | c     | c     | 0     |

Step ⑥ There is no final state unlike in a DFA.

If start state is given take it as start

## State Transition diagram.



Step ① : Equations

Step ② : Transition Table

Step ③ : Rewrite the Transition Table  
with:

Step ④ : Excitation Table

Step ⑤ : Assigning States &  
writing State Table

Step ⑥ : State Transition Diagram.

Q. For the above DFA :

If  $n = 11110011$   
Starting seq. a

what is Q/P seq  $Y = ?$

Ans .

a :  $i/p = 1 \quad o/p = 1$  state : d

d :  $i/p = 1 \quad o/p = 1$  state : c

c :  $i/p = 1 \quad o/p = 1$  state : a

a :  $i/p = 1 \quad o/p = 1$  state : d

d :  $i/p = 0 \quad o/p = 0$  state : c

~~a~~ :  $i/p = 0 \quad o/p = 0$  state : a

a :  $i/p = 1 \quad o/p = 1$  state : d

d :  $i/p = 1 \quad o/p = 1$  state : c

If  $x = 11110011$

$\therefore y = 11110011$

Intermediate State sequence  
= dcadcadcd