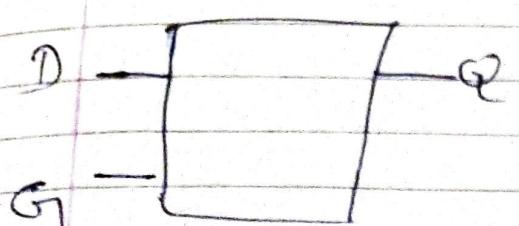
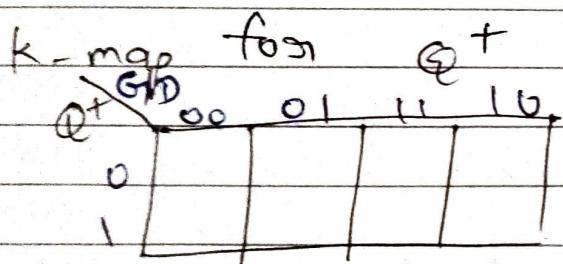


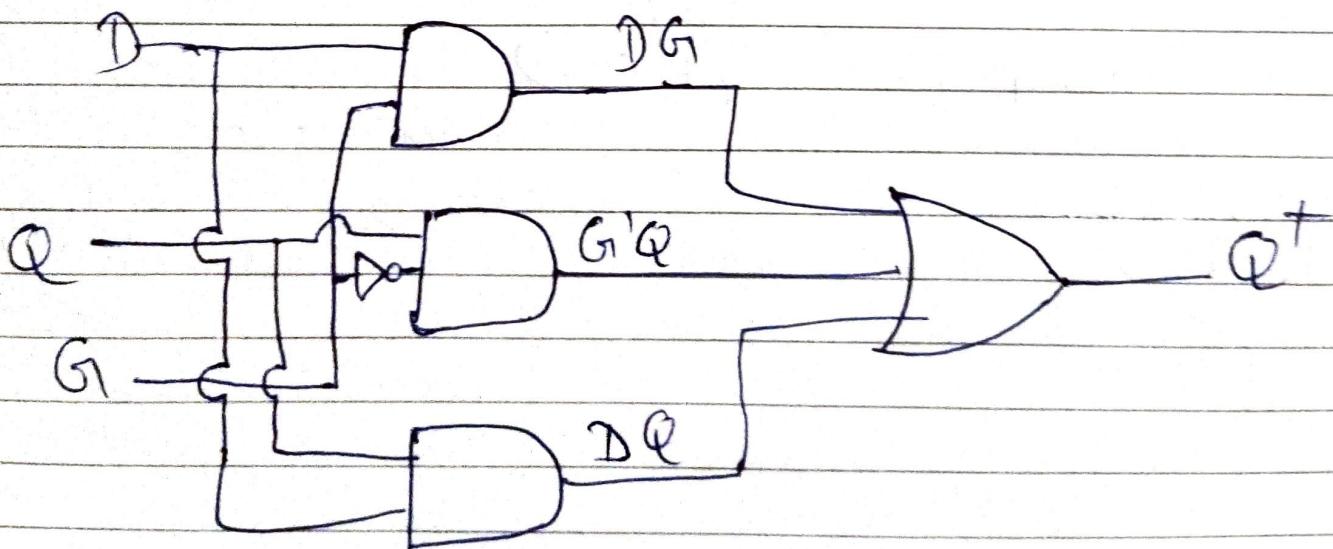
\* Transparent D Latch  
(Gated Latch) ~



D	G1	Q	$Q^+(N.S)$
0	0	0	0 } follows
0	0	1	1 } Q
0	1	0	0 }
0	1	1	1 }
1	0	0	0 } follows
1	0	1	1 } D.
1	1	0	1 }
1	1	1	1 }



$$\Rightarrow Q^+ = D G_1 + G_1' Q + D Q$$

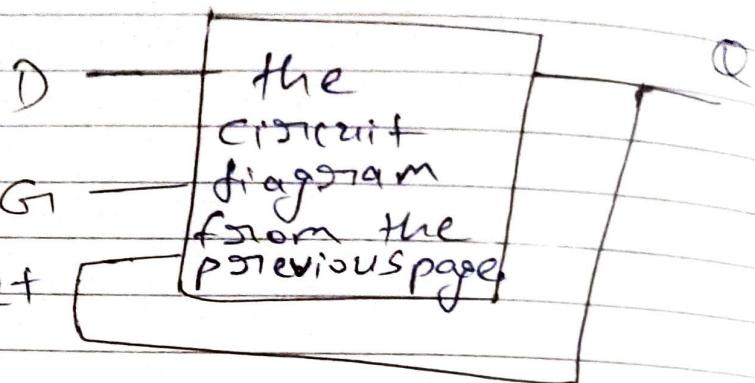


\* In terms of  $Q$  (as  $Q^t$  is Next state) and  $Q$  is available from the flipflop.



D flipflop.

$$Q^t = D G_1 + G_1 Q$$



after at  $Q \rightarrow Q^t$

implementation of transparent  
D latch

\* State Machine  $\rightarrow$  Synchronous & Asynchronous

$\downarrow$   
Mealy  
output =  $f(i_{pt}, c.s.)$

$\downarrow$   
current state

Moore

output =  $f(state)$

\* BCD to Excess 3 (code converter)

# State table.

PAGE NO. 5  
DATE

<u>X input (BCD)</u>	<u>Excess 3 case</u>
t <sub>3</sub> t <sub>2</sub> t <sub>1</sub> t <sub>0</sub>	t <sub>3</sub> t <sub>2</sub> t <sub>1</sub> t <sub>0</sub>
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 1 0 0

I/O table,

- ↳ Design a sequential state-Machine
- 1) write I/O table
- 2) State diagram.
- 3) state assignment

$$\begin{array}{r}
 \text{i/p} \quad 0100 \rightarrow \text{BCD} \\
 \text{add} \quad 0011 \\
 \hline
 \text{o/p} \quad \underline{0111} \rightarrow \text{Excess 3}
 \end{array}
 \qquad
 \begin{array}{r}
 \text{add} \quad 0011 \\
 \hline
 \text{o/p} \quad \underline{1000}
 \end{array}$$

- ↳ Initial state S<sub>0</sub>, z=0 & carry=0
- ↳ First bit arrives, we need to add '1' to this bit.

at t<sub>0</sub> we add '1' to the LSB.       $\checkmark$  S<sub>2</sub>

if X=0 (0+1) output z=1 (no carry)

if X=1 (1+1) output z=0 (carry: 1)

↳ z/o leads to S<sub>2</sub>

(i) at  $t_2$ : if there is no carry from first addition ( $s_1$ )

state  $s_{22}$  if  $x=0$  gives  $z = 0 + 2 + 0 = 2$

$x=1 \Rightarrow z = 2 + 2 + 0 = 4$

$x=2 \Rightarrow z = 1 + 2 + 0 = 3$

$x=0 \leftarrow$  leads to  $s_2$

(ii) if there is a carry state from first addition ( $s_1$ )

~~22.~~  $x=0$  gives  $z = 0 + 1 + 1 = 2$

$x=1$  gives  $z = 1 + 1 + 1 = 3$

$x=2$  gives  $z = 1 + 1 + 1 = 4$

$x=0 \leftarrow$  leads to  $s_1$

$x=1 \leftarrow$  have to add "1"

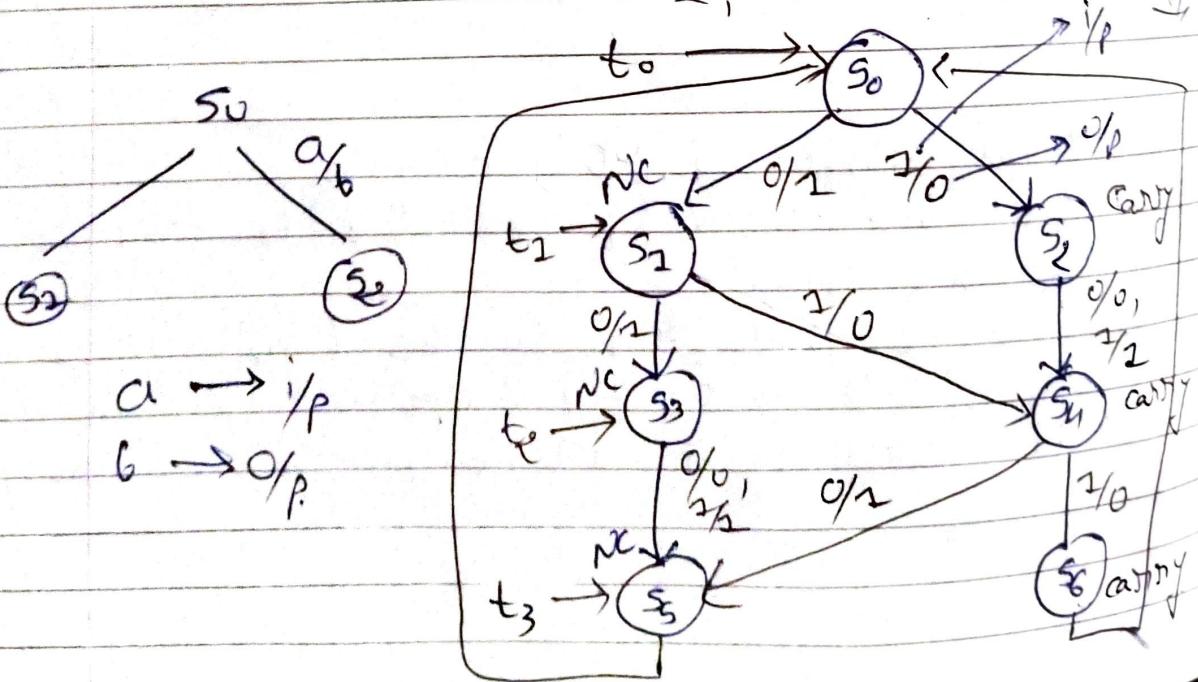
$x=2 \leftarrow$  carry : 2

$x=0 \leftarrow$  carry : 1

$x=1 \leftarrow$  carry : 1

$x=2 \leftarrow$  carry : 2

so on for  $s_2, s_3$

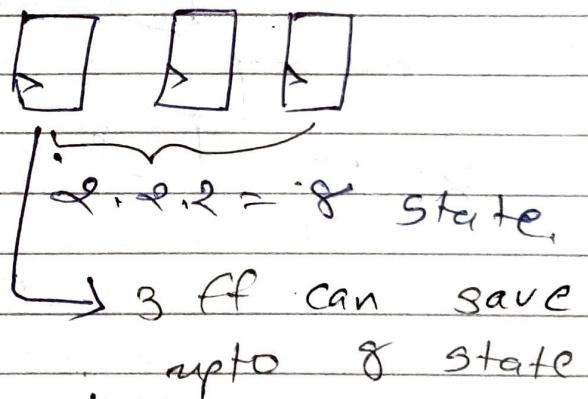
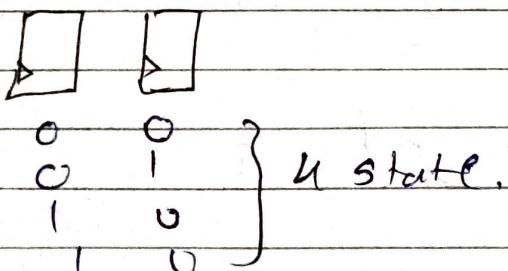


from state diagram to state table

Present state.	Next state.		Z (output)	
	$x=0$	$x=1$	$x=0$	$x=1$
$s_0$	$s_1$	$s_2$	1	0
$s_1$	$s_3$	$s_4$	1	0
$s_2$	$s_4$	$s_4$	0	1
$s_3$	$s_5$	$s_5$	0	1
$s_4$	$s_5$	$s_6$	1	0
$s_5$	$s_6$	$s_0$	0	1
$s_6$	$s_0$	-	1	-

7 state  $\Rightarrow s_0 : 3 \text{ ff}'s$

Each ff/register can save  $\leq$  state  $\approx \frac{1}{2}$



we have to encode this state ( $s_0, s_1, s_2, s_3$ ) to  $b_2 b_1 b_0$ .

$b_2 = 0/011$

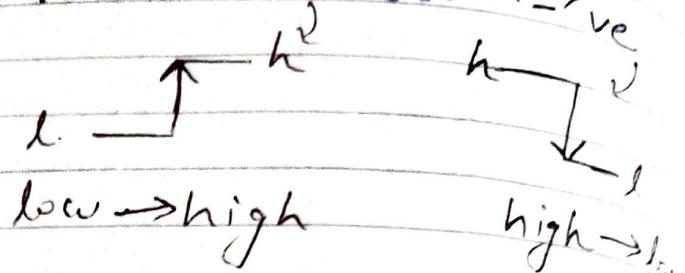
# # Lecture :- 4

## (Revision)

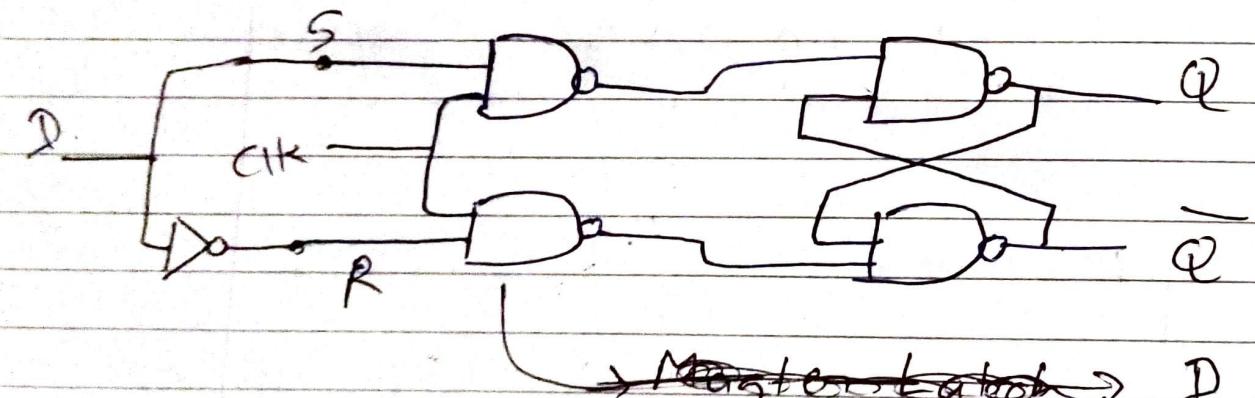
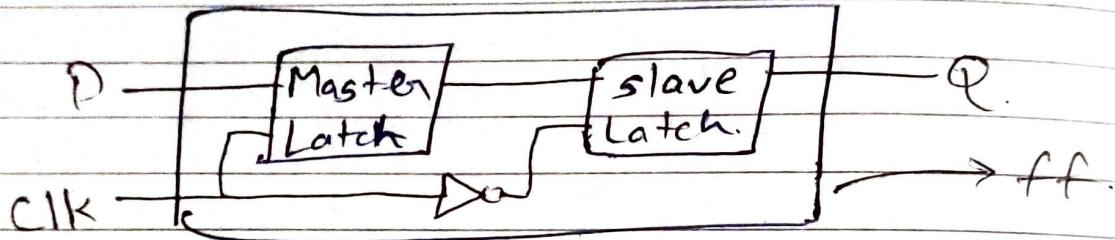
8/09

clocking Event :-

clk transition : +ve or -ve



↳ ff is combination of M. Latch + S. Latch



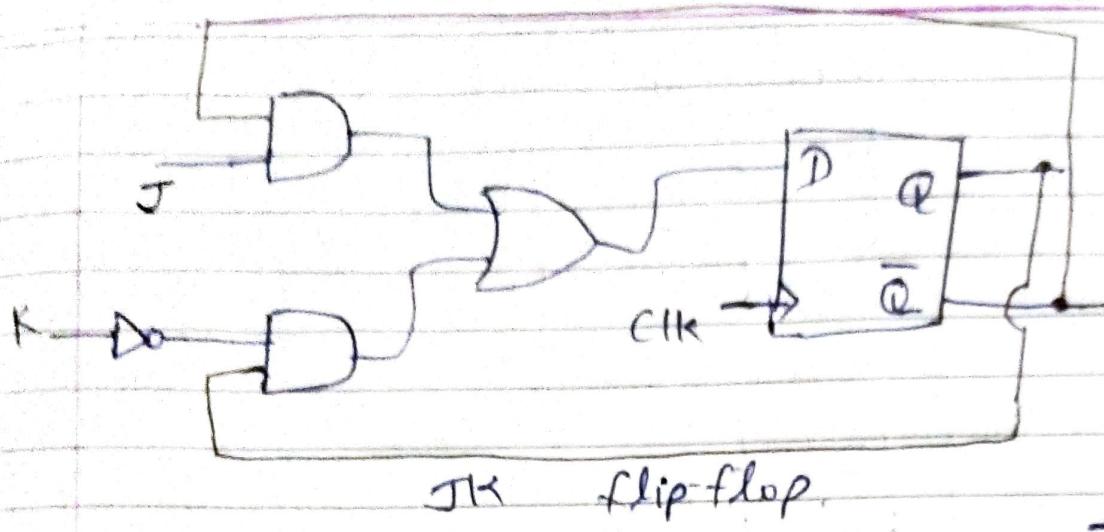
~~Master Latch~~ → D flip flop

D	CLK	Q
0	↑	0
1	↑	1

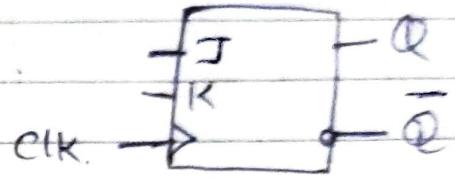
D ff

J	K	CLK	Q	Q'
0	0	↑	Q <sub>0</sub>	Q' <sub>0</sub>
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	Toggle	

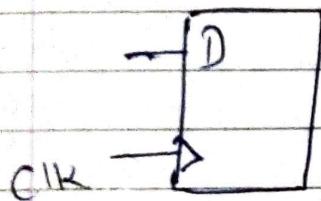
JK ff



JK flip-flop.



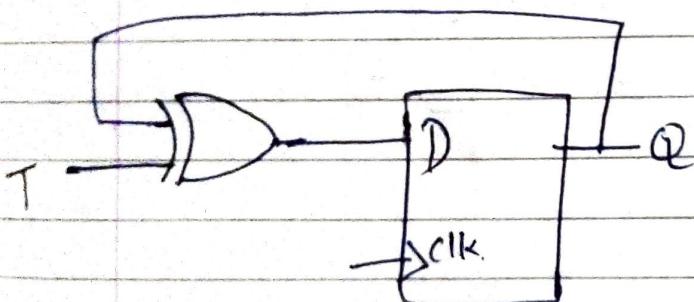
\* T flip-flop ~



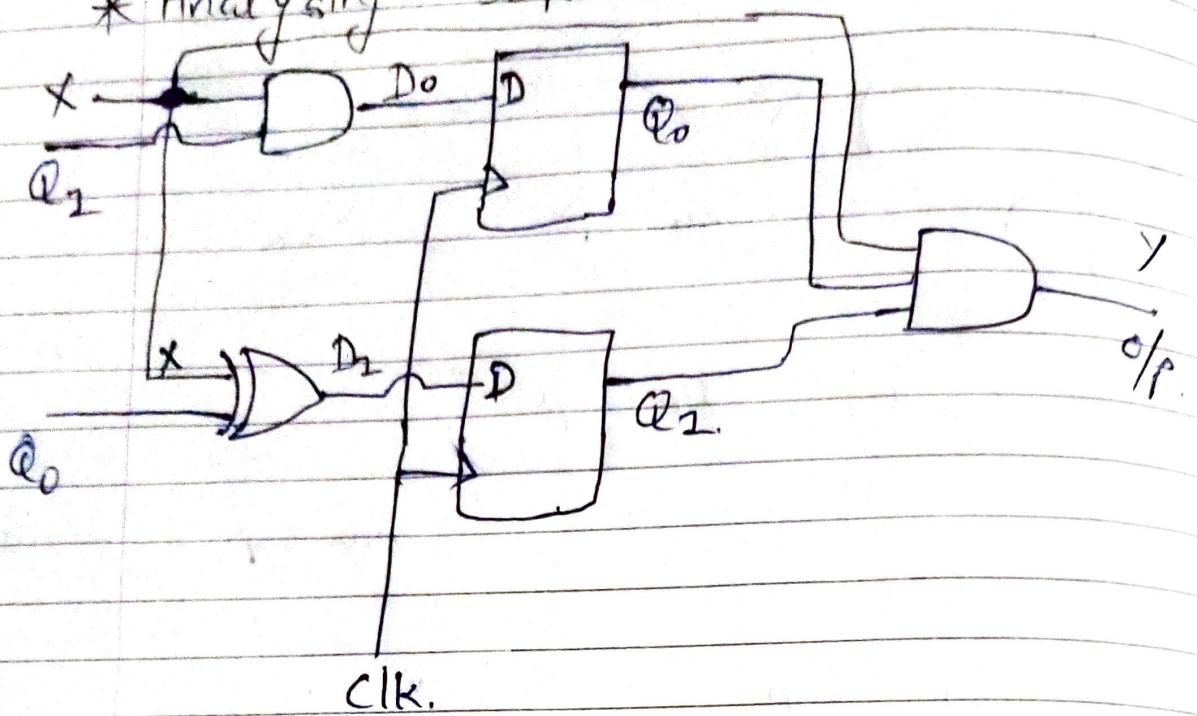
T	clk	Q	$Q'$
0	↑	$Q_0$	$Q'_0$
1	↑	$Q'_0$	$Q_0$

Toggle

$T = 0 \Rightarrow$  keep the current state  
 $T = 1 \Rightarrow$  inversion happens  
 $(1 \rightarrow 0 \text{ or } 0 \rightarrow 1)$



\* Analysing sequential ckt.



Equation ~

$$Y(t) = X(t) \cdot Q_0(t) \cdot Q_1(t)$$

$$Q_0(t+1) = D_0(t) = X_0(t) \cdot Q_2(t)$$

$$Q_1(t+1) = D_1(t) = \underline{X(t)} + \underline{Q_0(t)}$$

① State Equation

② State table ~ Sequence of I/P, O/P, etc

PS (present state) - states current value of flip-flop

NS (next state) - indicates the states after the next rising clk edge.

Output : output value at current clk edge

In the state table.

- all possible ip combination will be defined
- all possible state combination will be defined.

ip : x 1/2 0	PS	NS		Output	
		x=0	x=1	x=0	x=1
	S <sub>0</sub>	S <sub>0</sub>	S <sub>1</sub>	0	0
	S <sub>1</sub>	S <sub>2</sub>	S <sub>2</sub>	0	0
	S <sub>2</sub>	S <sub>0</sub>	S <sub>3</sub>	0	0
	S <sub>3</sub>	S <sub>2</sub>	S <sub>3</sub>	0	1

State encoding :-

$$S_0 = 00$$

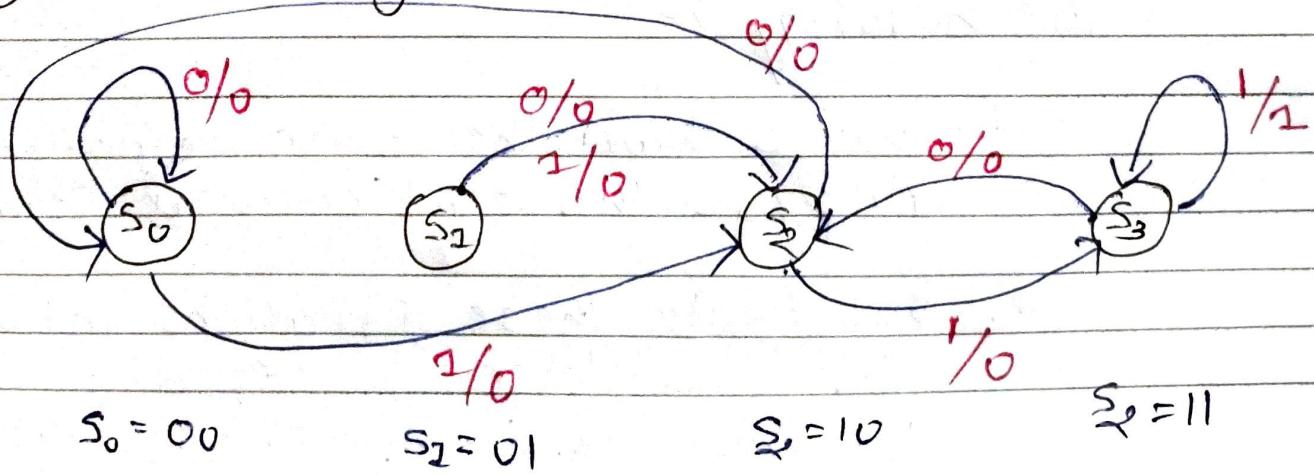
Encode, Enumerate.

$$S_1 = 01$$

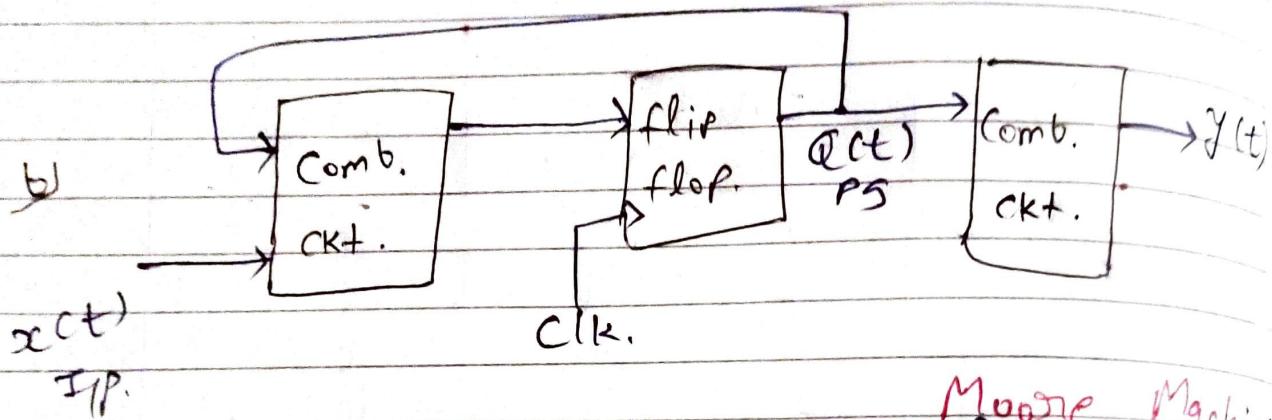
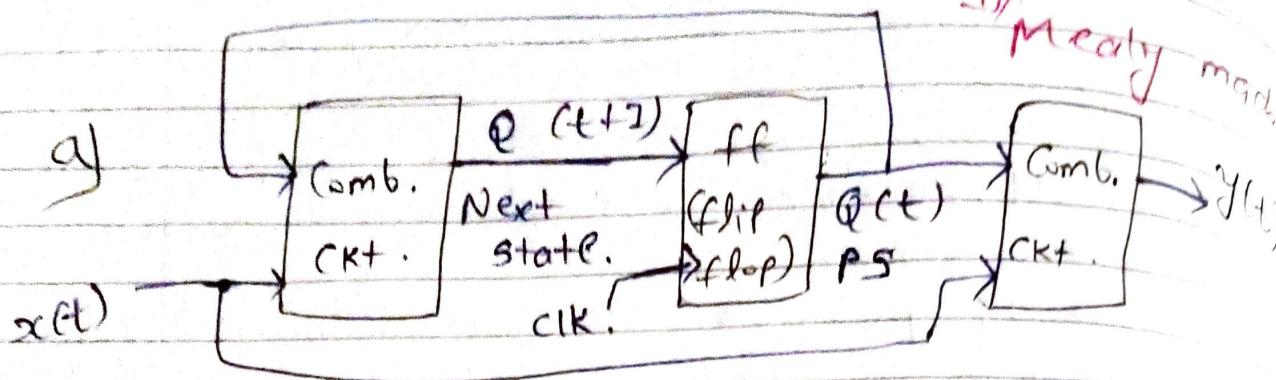
$$S_2 = 10$$

$$S_3 = 11$$

③ state diagram n ip/op



$\text{Output} = f(\text{state}, \text{Present I/P})$



$\text{Output} = f(\text{state only})$

for moore machine.

↳ Since the outputs only depends on the ps the outputs cannot change during the clk pulse if the ip variable change.

\* Methodology ~

1. Write down the state equation  
Analyse the combinatorial ckt. (of)

2. Substitute these equations into ff. table equa  
for the transition. ex  $Q_1(t+1) = f(01)$

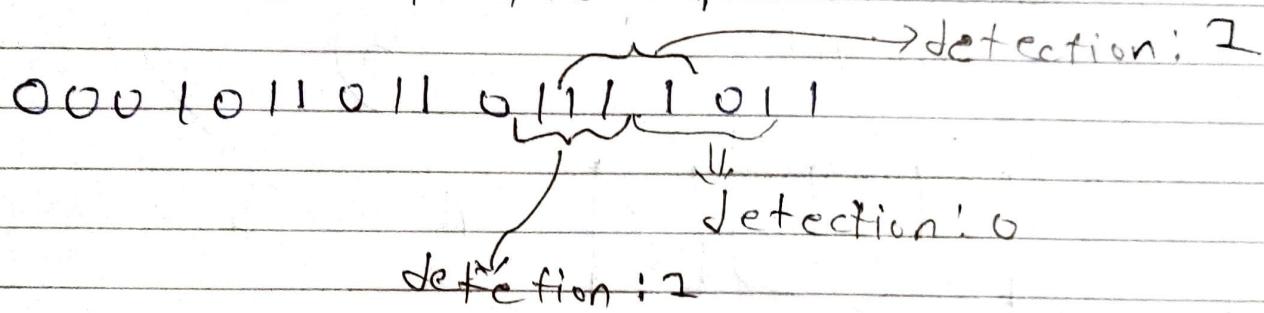
3. Find the O/P equations.

$$Y = f'(Q, i/p)$$

4. Construct a state transition table, O/P table from the above transition and O/P equation (3).

5.

Example: Detection of 3 consecutive 1 (in the input.)



$S_0$ : zero 1 detected.  $\xrightarrow{0}$

$S_1$ : one 1 detected.  $\xrightarrow{0}$

$S_2$ : two 1 detected.  $\xrightarrow{0}$

$S_3$ : three 1 detected.  $\xrightarrow{1}$

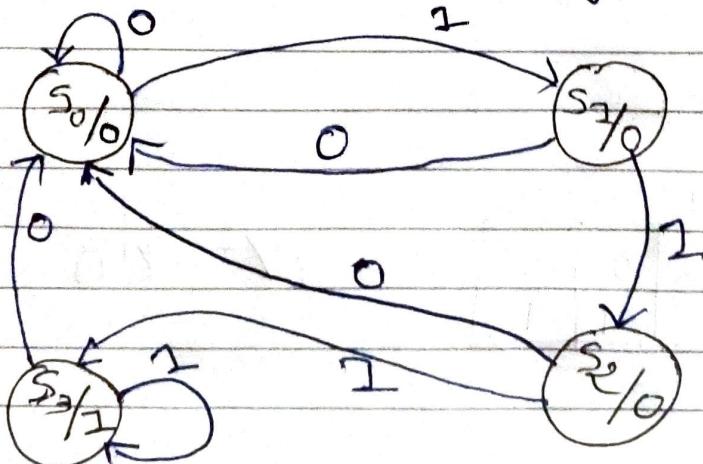
O/P Output = 1; if

3 consecutive 1s

(2) else

= 0 else

②  
State  
diagram.



O/P is  
only depend  
on  
states.

$\Rightarrow$  Moore  
machine.

$$S_2 = \frac{1}{2}, S_1 = 01, S_0 = 10, S_3 = 11$$

(e) state table

PS A B	input X	NS $A^+ B^+$	Output Y	
			0 0	0 1
S <sub>0</sub> 0 0	0	0 0	(S <sub>0</sub> )	0
S <sub>0</sub> 0 0	1	0 1	(S <sub>2</sub> )	0
S <sub>2</sub> 0 1	0	0 0	(S <sub>0</sub> )	0
S <sub>2</sub> 0 1	1	1 0	(S <sub>2</sub> )	0
S <sub>0</sub> 1 0	0	0 0	(S <sub>0</sub> )	0
S <sub>0</sub> 1 0	1	1 1	(S <sub>3</sub> )	0
S <sub>3</sub> 1 1	0	0 0	(S <sub>0</sub> )	1
S <sub>3</sub> 1 1	1	1 1	(S <sub>3</sub> )	1

for A <sup>+</sup>	
A	$\cancel{B} X$
0	00 01 11 10
1	0 0 1 1 0

③

state equation.

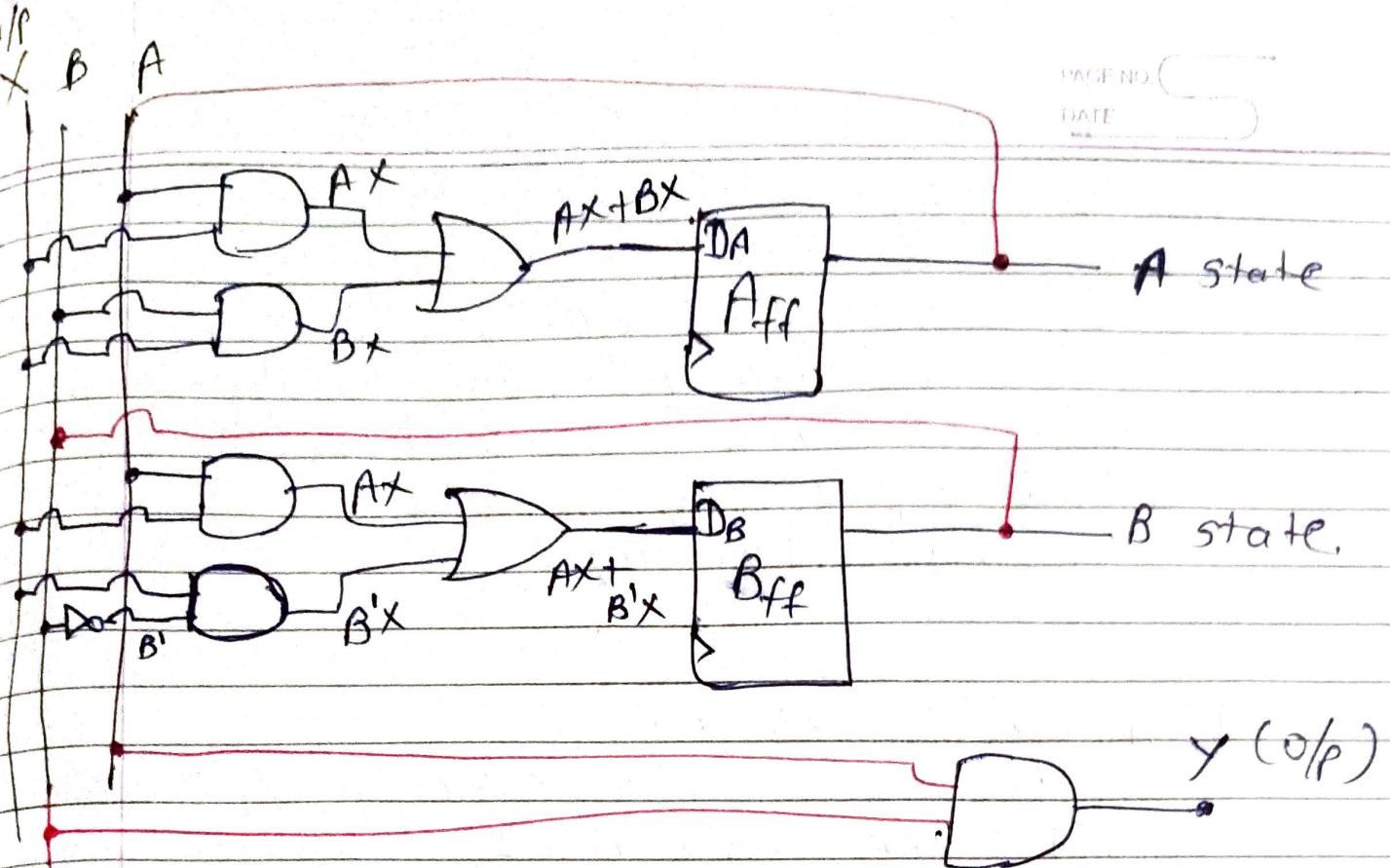
$$A^+ \text{ or } D_A = B' X + A X$$

for B <sup>+</sup>	
A	$\cancel{B} X$
0	00 01 11 10
1	0 0 1 0 0

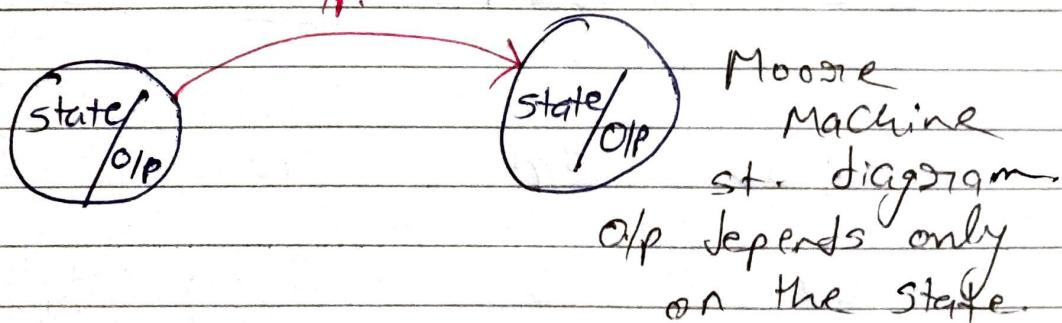
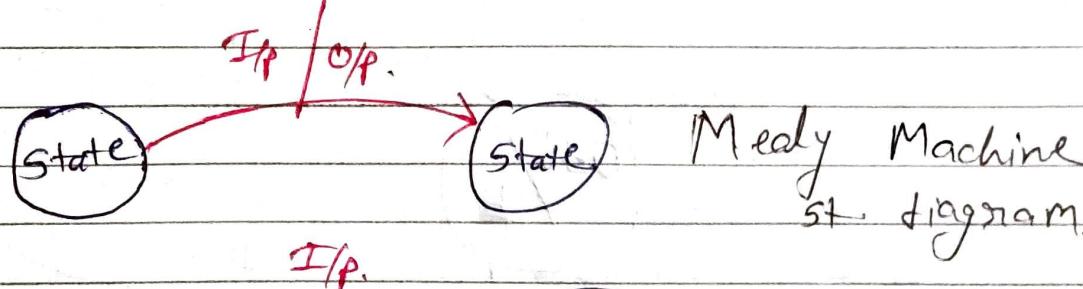
$$B^+ \text{ or } D_B = B' X + A X$$

for O/P	
A	$\cancel{B} X$
0	00 01 11 10
1	0 0 0 0 0

$$Y = AB$$



Moore Machine,  
Consecutive 3 1's Detection  
Machine.



# # Lecture 7

10/09

## \* Designing of Finite state Machines FSMs

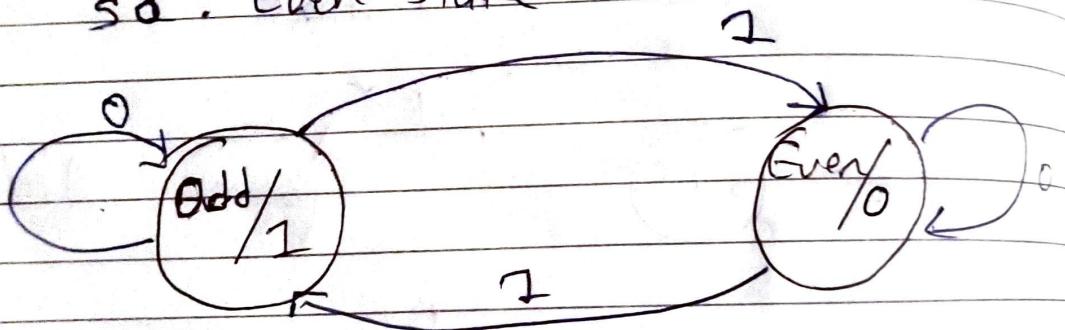
- 1) Design the problem with simple logic.
- 2) Encode binary value to the states.
- 3) State table & state diagram.
- 4) K-maps for expression.
- 5) Logic diagram.

### \* Odd Parity checker:

if no. of ones in binary representation is odd then return '1' else return '0'

S<sub>0</sub>: Odd state.

S<sub>1</sub>: Even state



PS	If	NS	O/P
0	Even	0	Even (0)
0	Even	1	Odd (1)
1	Odd	0	Odd (1)
1	Odd	1	Even (0)

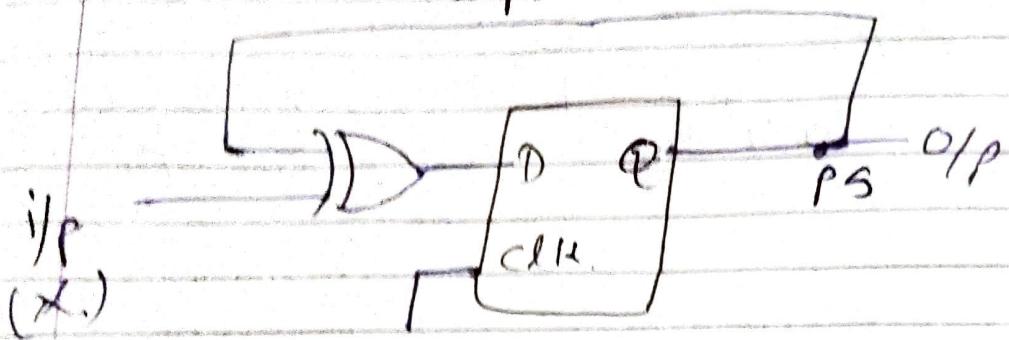
0	Even	0	Even (0)	0
0	Even	1	Odd (1)	0
1	Odd	0	Odd (1)	1
1	Odd	1	Even (0)	1

Even : 0  
Odd : 1 } Encode.

No. equation

$$NS = PS \oplus \text{input}$$

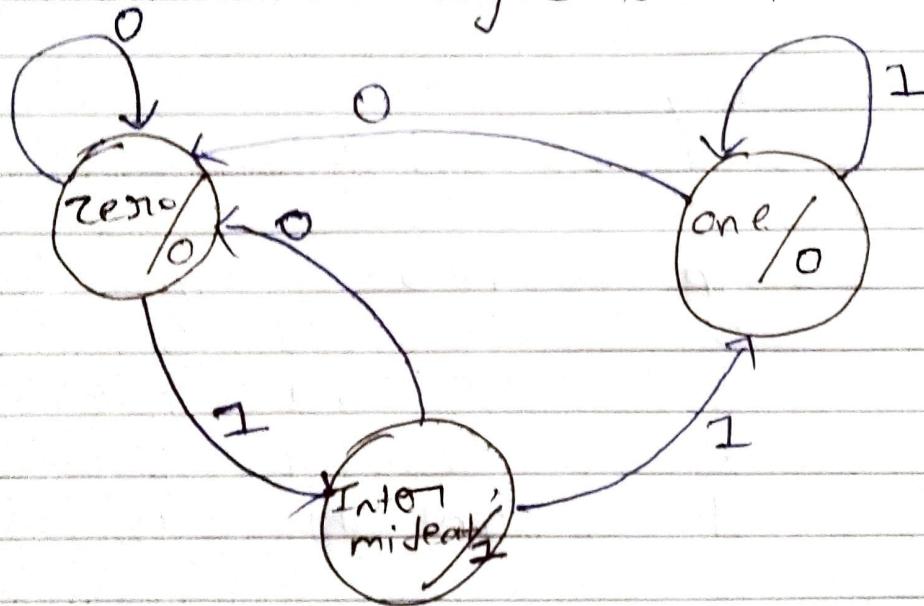
$$\text{output} = PS$$



\* Edge Detection  $\rightarrow$  Left to Right.

0000000100

$\rightarrow$  O/P is 1 if the i/p bit stream changes from 0 to 1



Moore

Encoding

zero (00)

int (01)

one (11)

A B

PS	I/P	NS	O/P
zero(00)	0	zero(00)	0
zero(00)	1	int (01)	0
int (01)	0	zero (00)	0
int (01)	1	one (11)	1
one (11)	0	zero (00)	0
one (11)	1	one (11)	0

$$PS \rightarrow PS_0$$

00

01

11

10

01

11

$$A = (I/p) \cdot PS_0$$

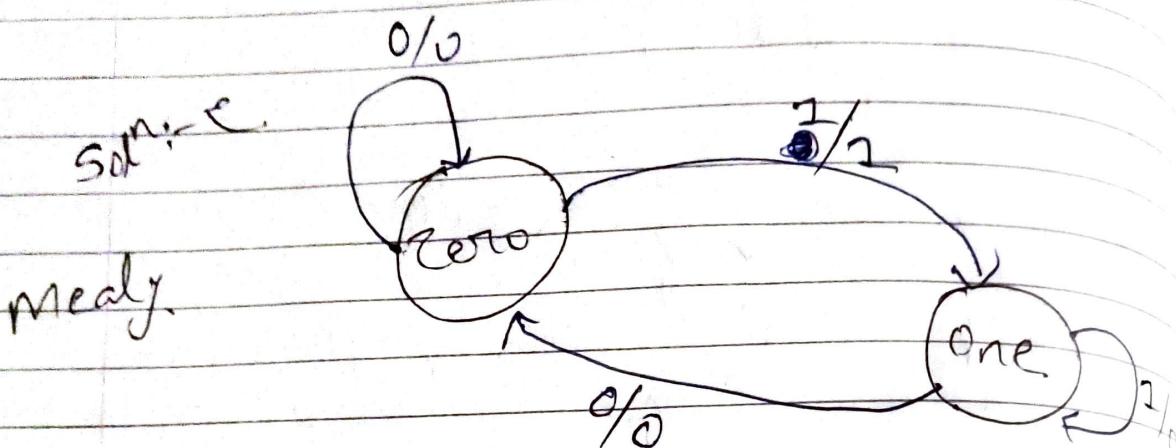
$$A = (I/p) \cdot PS_1$$

$$\beta = (I/p)$$

$$O/p = \bar{PS}_0 \cdot PS_1$$

$PS_0$  means  
0/p of 2<sup>st</sup> ff.  
 $PS_1$  means  
0/p of  
1st ff.

can derive  
from state  
using K-map



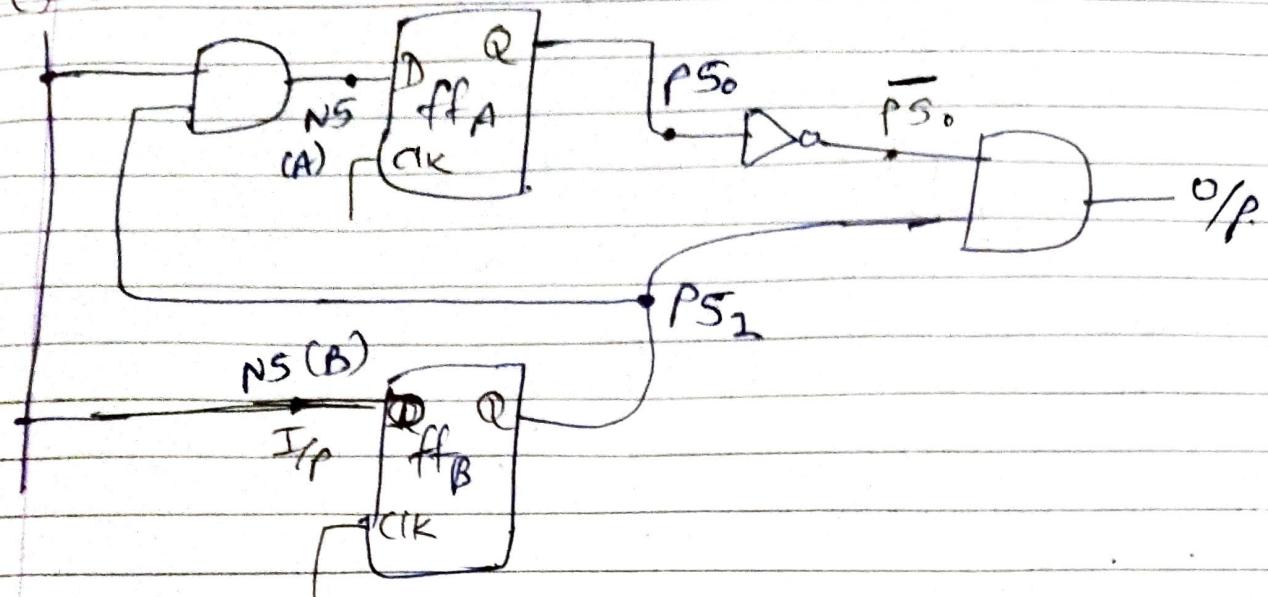
Encoding zero: 0  
one: 1

PS	I/p	NS	O/p
zero(0)	0	zero(0)	0
zero(0)	1	one(1)	1
one(1)	0	zero(0)	0
one(1)	1	one(1)	0

$$NS = I/p$$

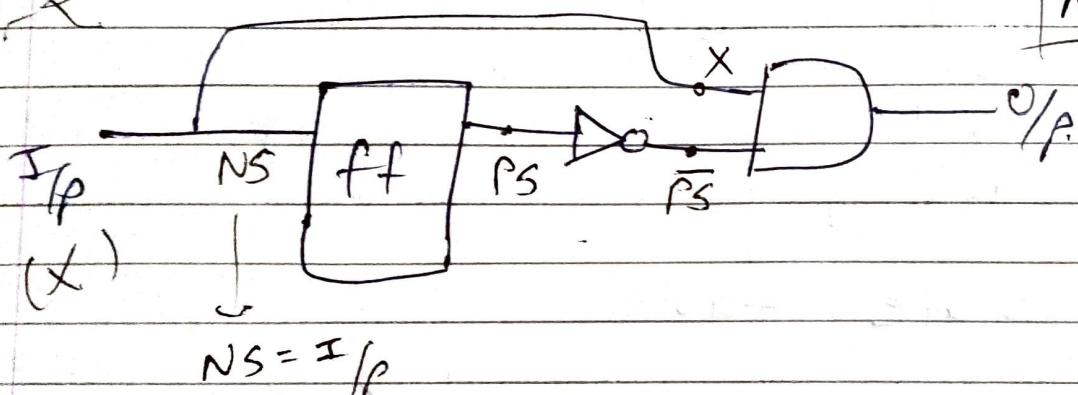
$$O/p = I/p \cdot \bar{PS}$$

Sol: 1 (Moore)



Sol: 2.

[Mealy]



\* State Reduction ~

↳ we will do a row matching in the state table.

↳ In two states if the  $\rightarrow$  states has same o/p &

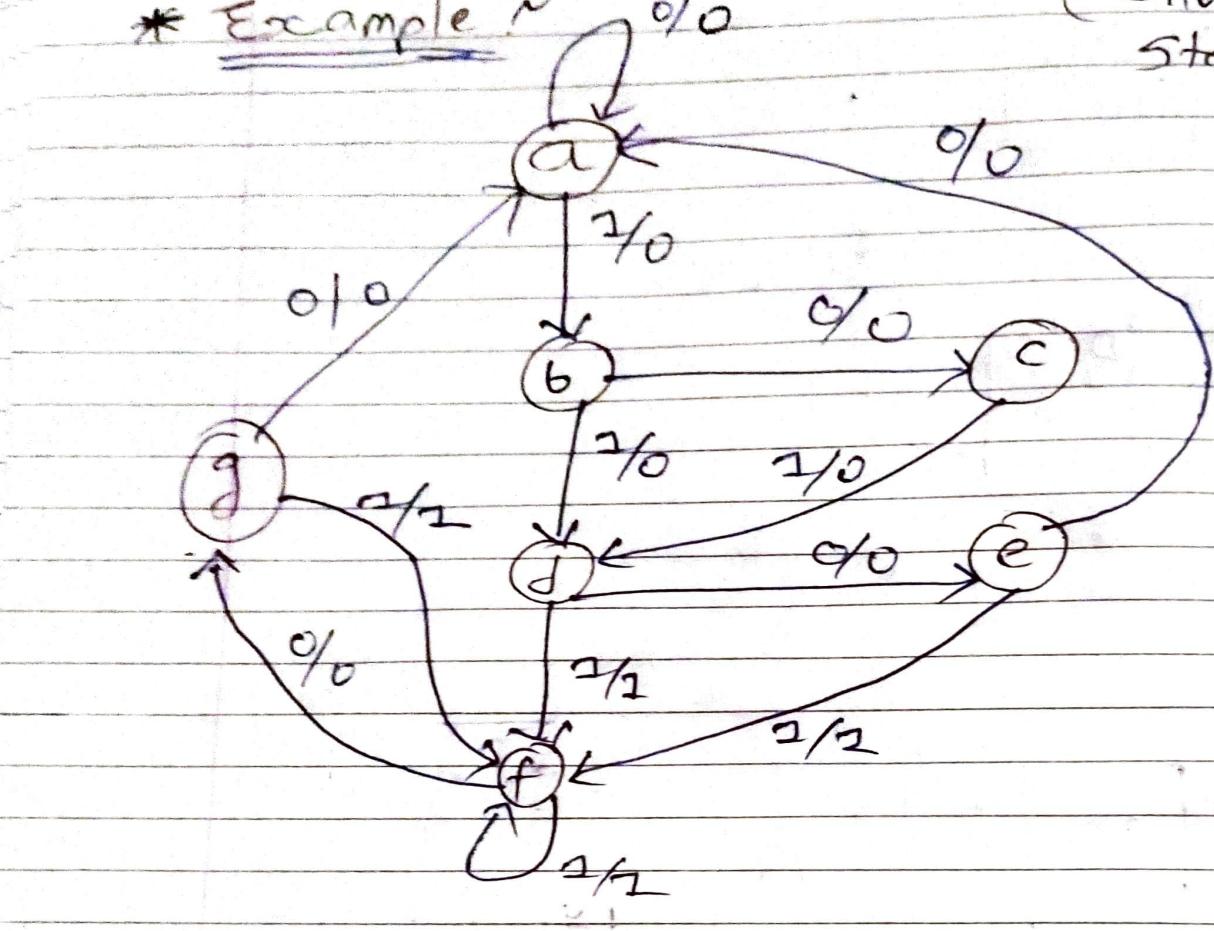
- they go to the same NS.
- or both self loop.
- or both to each other

They are EQUIVALENT.

\* Example

(Given)

State diagram.



State table

PS	NS		O/P	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	g	0	1
g	e	f	0	1
j	a	f	0	1

→ Redundant NS & O/P is same as for e.

now we have only 5 state  
a, b, c, d, e.

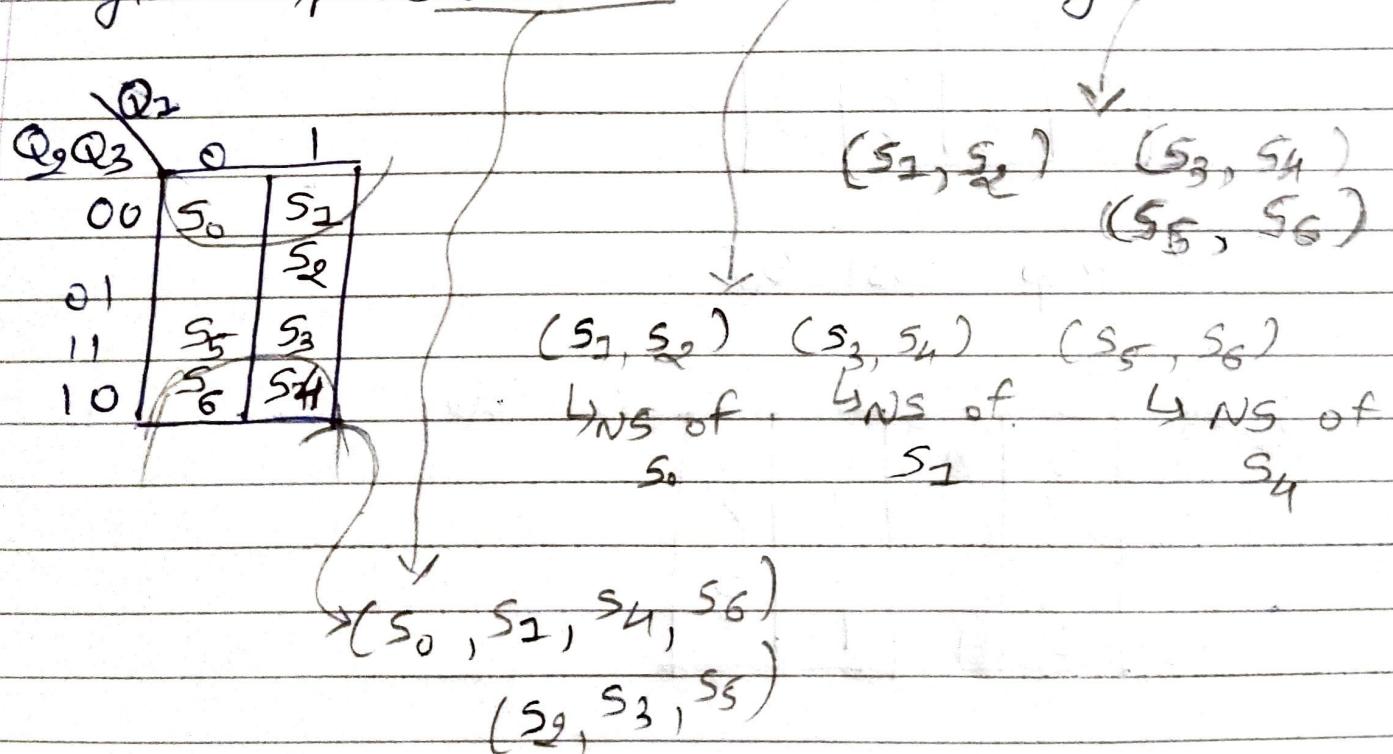
# Lecture ~6.

15/09

BCD to Excess 3 code converter ~  
(Continue)

State Encoding ~

- ↳ states which have the same NS for a given ip should be clubbed together.
- ↳ state which are the NS of the some state should be clubbed together.
- ↳ states which have the same O/p for a given ip should be clubbed together.



$$S_0 = 000$$

$$S_1 = 100$$

$$S_2 = 101$$

$$S_3 = 111$$

$$S_4 = 110$$

$$S_5 = 011$$

$$S_6 = 010$$

$S_6$	$S_5$	$S_4$	$S_3$	$S_2$	$S_1$	$S_0$	$X = 0$	$X = 1$	$Z$
000							$(S_1) 100$	$(S_2) 101$	1
100							$(S_3) 111$	$(S_4) 110$	1
011							$(S_4) 110$	$(S_5) 011$	0
111							$(S_5) 011$	$(S_6) 010$	0
110							$(S_6) 010$	$(S_0) 000$	1
010							$(S_0) 000$	-	0
010							$(S_0) 000$		1

K-map for  $\overline{Q_2} + Q_3 +$

$\overline{Q_2} \backslash Q_3$	00	01	11	10
00	1	1	1	1
01	x	1	1	x
11	0	0	0	0
10	0	0	0	x

$D_1 = Q_2^+ = \overline{Q}_2$

K-map for  $Q_2^+$

$\overline{Q_2} \backslash Q_3$	00	01	11	10
00	0	1	1	0
01	x	1	1	x
11	0	1	1	0
10	0	1	1	x

$D_2 = Q_2^+ = Q_2$

K-map for  $Q_3^+$

<del><math>X Q_2</math></del>	00	01	11	10
$Q_2 \ Q_3$	00	0	1	0
00	0	X	0	X
01	X	0	0	X
11	0	1	1	0
10	0	X	0	X

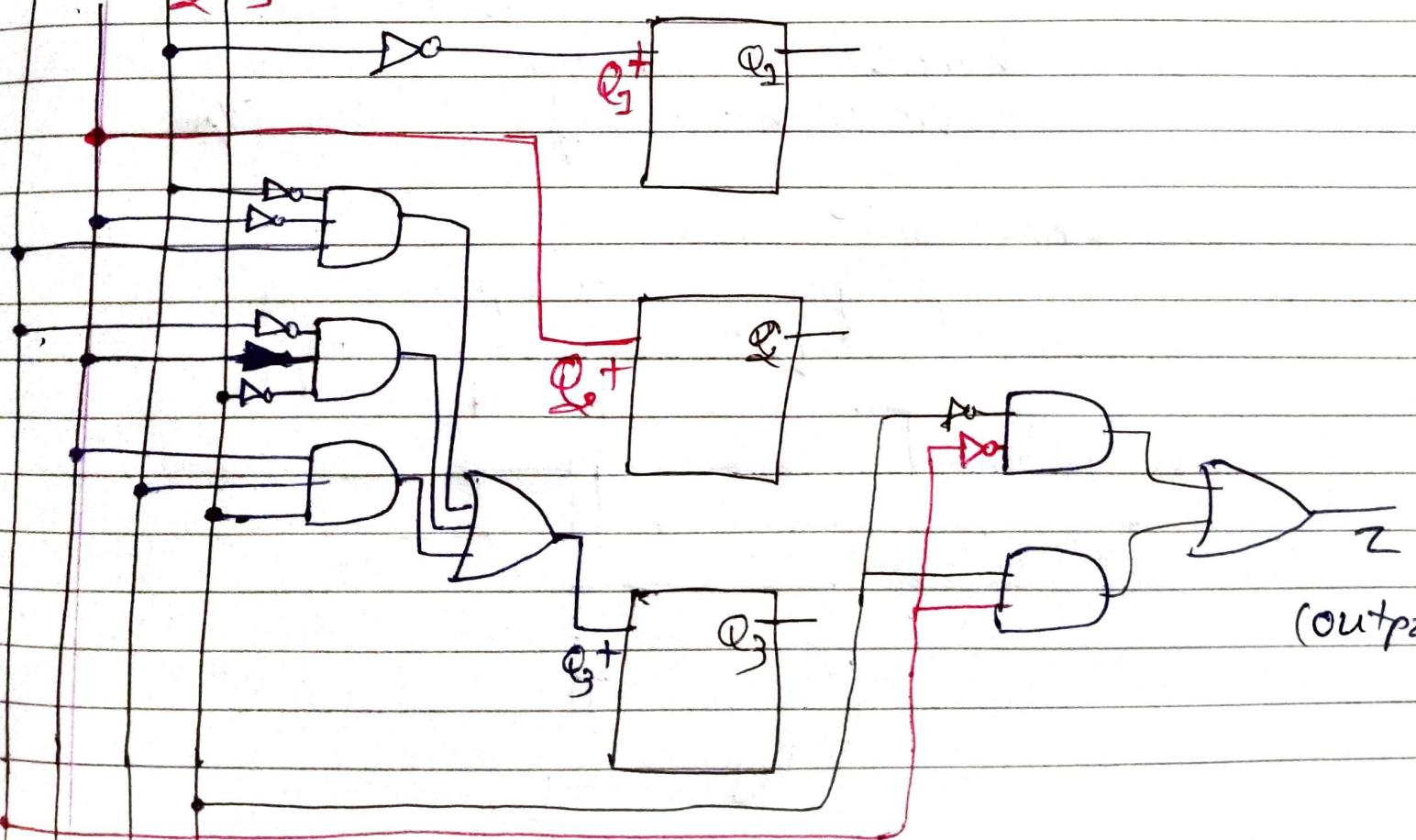
$$D_3 = Q_3^+ = Q_2' \times Q_1' + Q_2 Q_3 Q_1 + Q_3' \times' Q_2$$

K-map for output  $z^{(2)}$

<del><math>X Q_2</math></del>	00	01	11	10
$Q_2 \ Q_3$	00	1	1	0
00	1	X	1	X
01				
11			1	1
10	1	1	0	X

$$z = Q_2' x' + Q_2 x$$

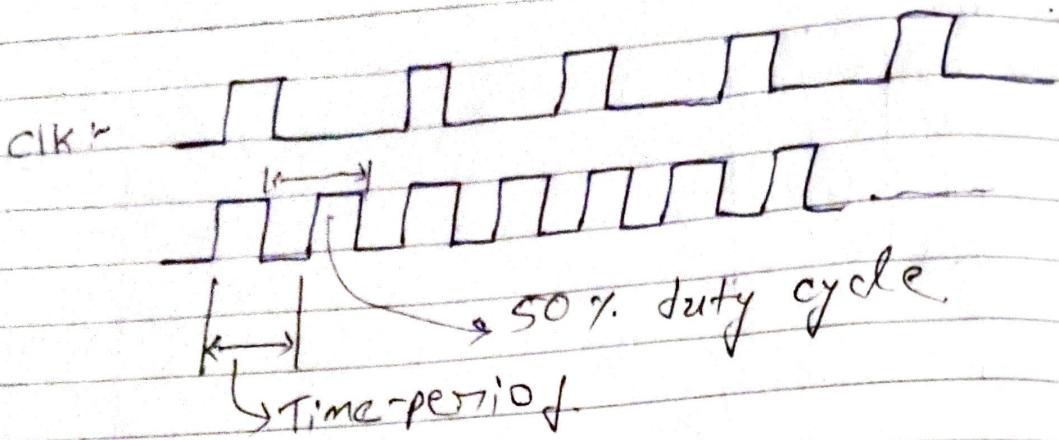
\*  $Q_2$  Code Converter Circuit:



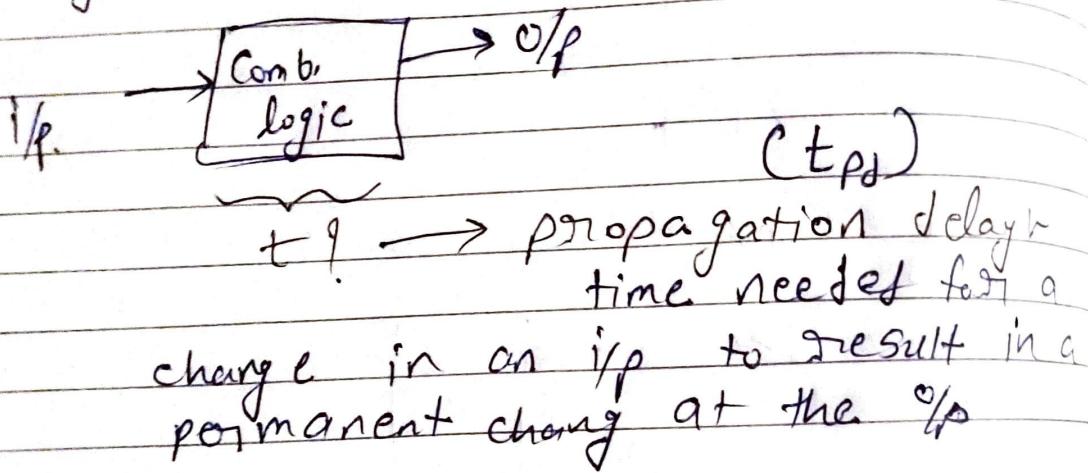
# # Lecture 17

17/09

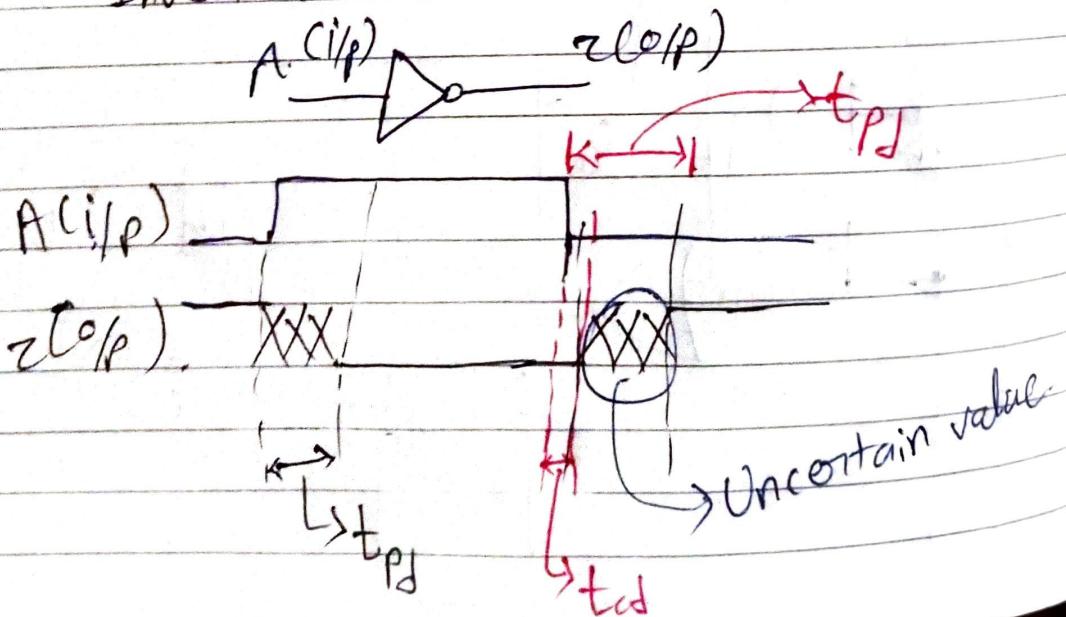
## \* Sequential ckt. Timing ~



↳ Timing parameters of the combinational logic ~



Inverter ~

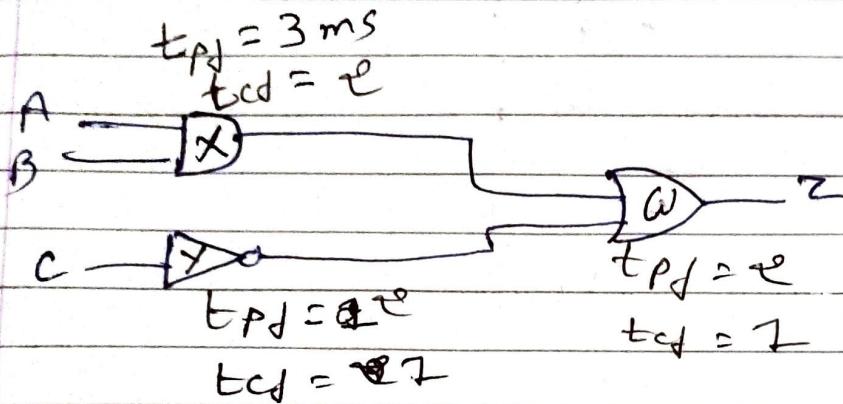


$t_{cd}$  ~ contamination delay :-

Time on delay indicate the amount of time needed for a change in a logic input to result in an initial change at an output.

$$t_{cd} < t_{pd}$$

↳ Combinational logic delays  $\rightarrow$  always additive.



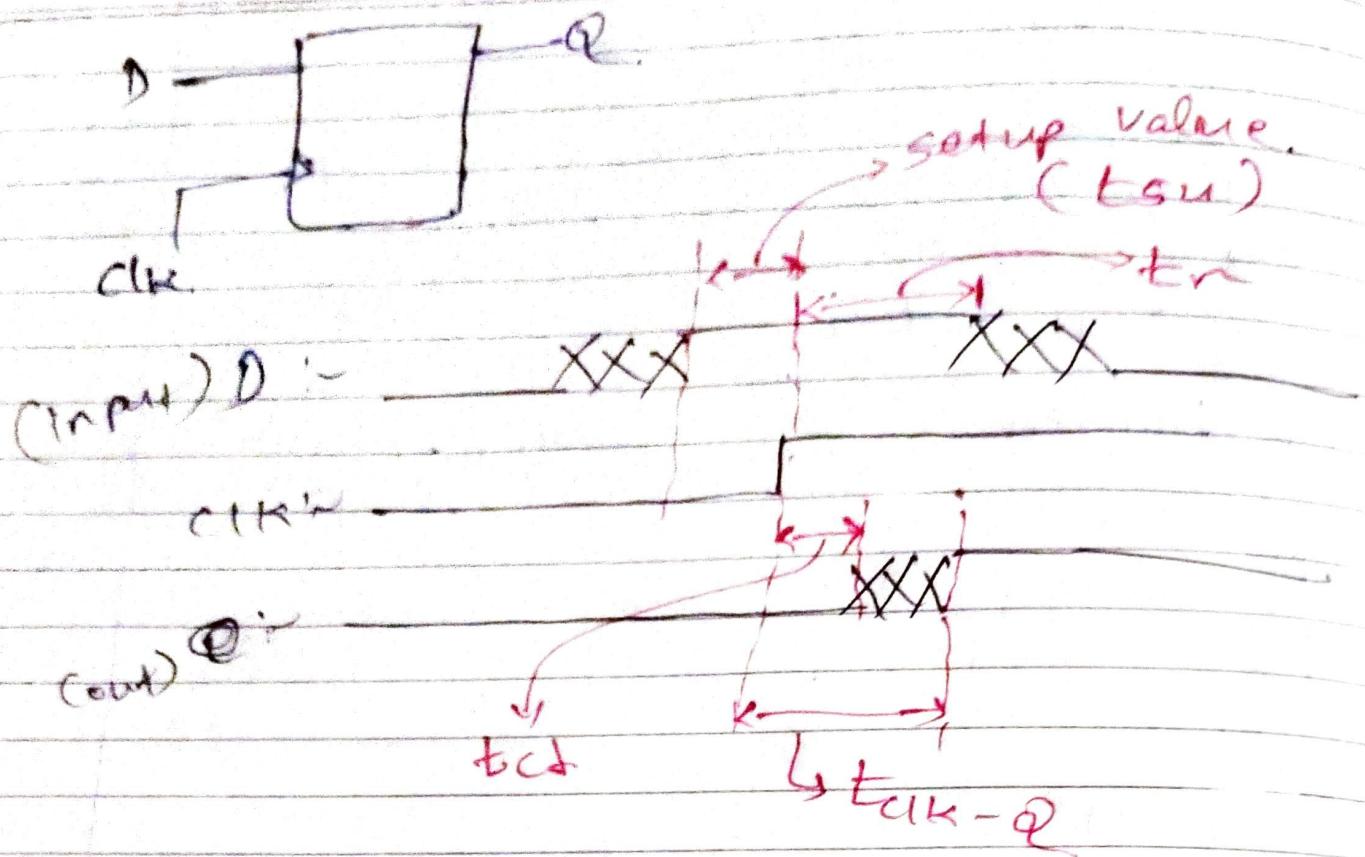
$$\begin{aligned} X \cdot \omega &\sim t_{pd} \text{ of } X + t_{pd} \text{ of } \omega \\ &= 3 + \tau \rightarrow t_{cd} = \tau + 2\tau = 3\tau \\ &\sim 5 \text{ ms} \end{aligned}$$

$$\begin{aligned} Y \cdot \omega &\sim t_{pd} \text{ of } Y + t_{pd} \text{ of } \omega \\ &= 2 + \tau \\ &= 4 \text{ ms. } \rightarrow t_{cd} = 2 + \tau = 2\tau. \end{aligned}$$

↳ Sequential ckt. :-

propagation delay :-  ~~$t_{clk-Q}$~~

time on value indicates the time needed for change in the ff clk input (rising or falling edge) to result in a permanent change in the ff output ( $Q$ ).

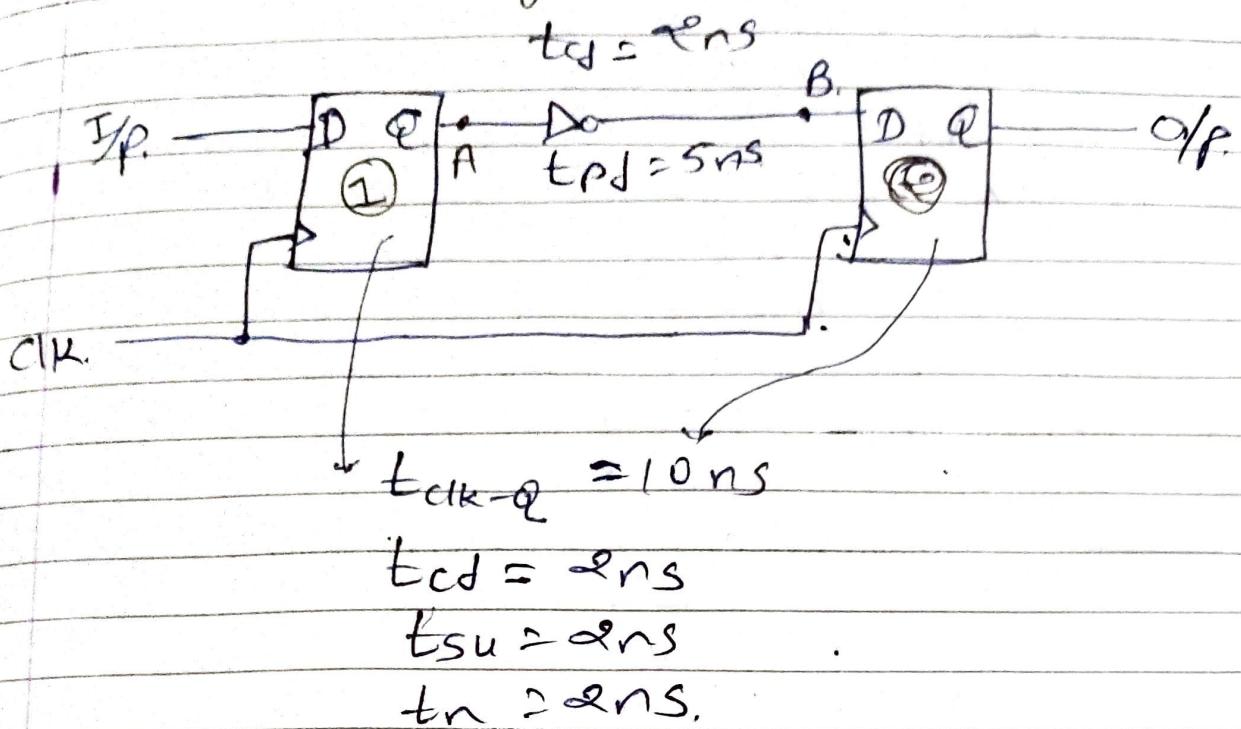


Contamination delay  $\downarrow$  Value indicates the amount of time needed for a change in the ff clk input to result in the initial change at the ff output Q.

Set up time  $\leftarrow$  Value indicates the amount of time before the clock edge that data input D must be "STABL".

Hold time  $\nwarrow$  Value indicates the amount of time after tr the clock edge the data input D must be "stable".

\* Determining maximum clock freq.



2. for ff<sub>1</sub> at  $f_{CLK}$  a valid o/p. appears at A after  $t_{CK-Q} = 10\text{ ns}$ .

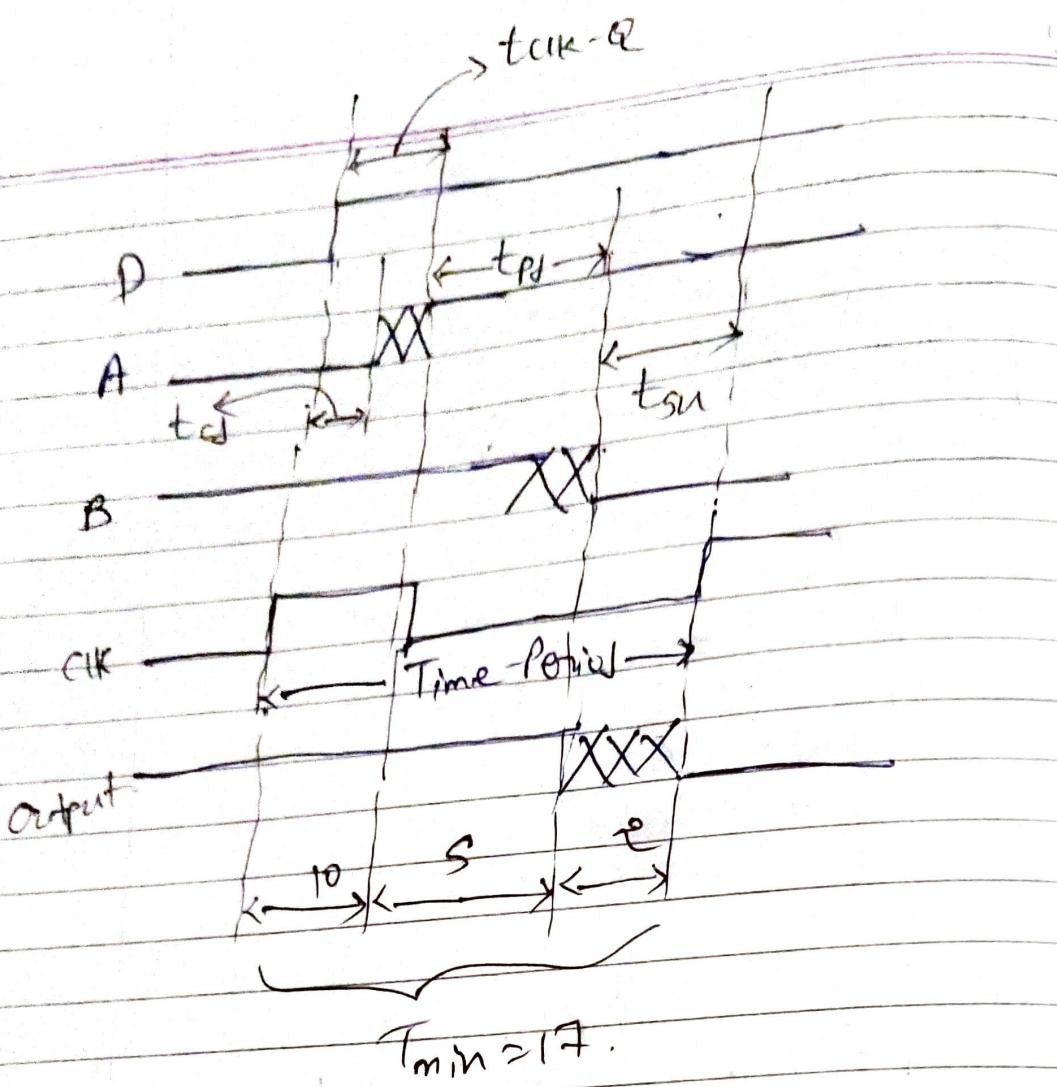
3. A valid o/p. appears at B after  $t_{PD} = 5\text{ ns}$  (for inverter.)

3. Signal B is clocked on the  $f_{CLK}$ .  
Condition: Signal must arrive at least  $t_{SU} = \infty\text{ ns}$  before the  $f_{CLK}$

for max. freq.

$$\begin{aligned} T_{min} &= t_{CK-Q}(1) + t_{PD}(\text{inverter}) + \\ &\quad t_{SU}(\infty) \\ &= 10 + 5 + \infty \\ &\approx 17\text{ ns} \end{aligned}$$

$$f_{max} = \frac{10^9}{17} \text{ Hz}$$



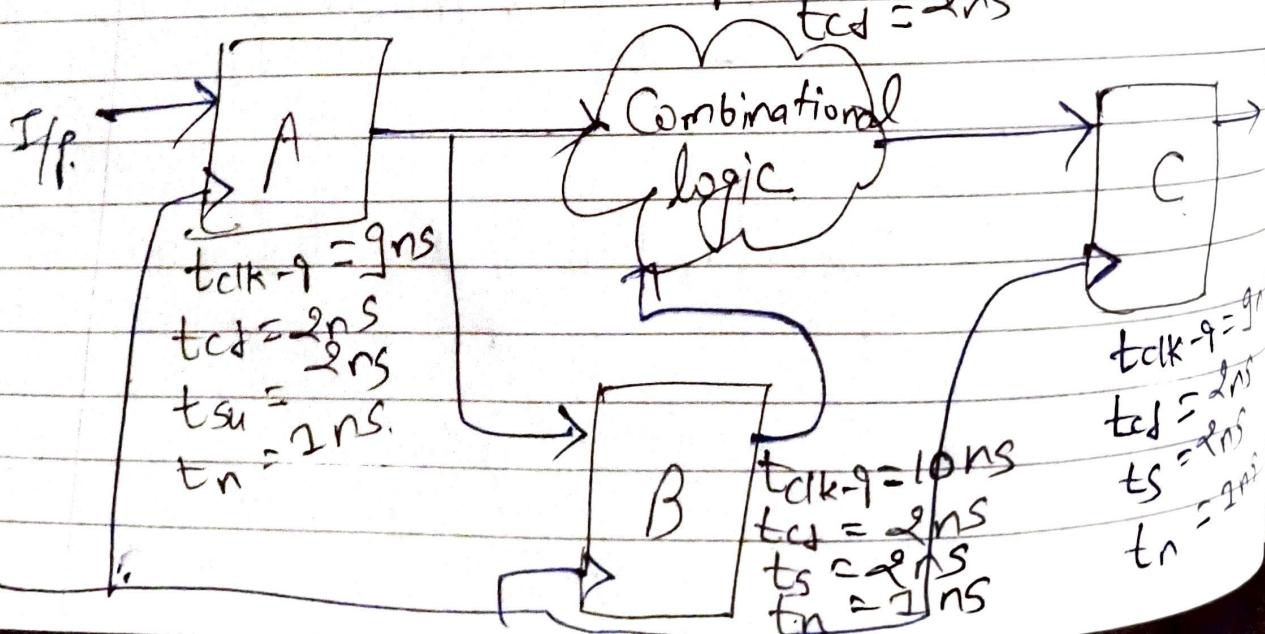
## # Lecture :- 18

22/09

\* Determining the clock freq. of sequential ckt.

$$t_{PD} = 4 \text{ ns}$$

$$t_{CD} = 2 \text{ ns}$$



↳ take into account both setup and hold time.

$$A \rightarrow B \quad B \rightarrow C \quad + \quad A \rightarrow C$$

$$T_{AB} = t_{CK-Q}(A) + t_{SU}(B) = 9 + 2 = 11 \text{ ns}$$

$$T_{AC} = t_{CK-Q}(A) + t_{PD}(cc) + t_{SU}(C) = \\ 9 + 4 + 2 = 15 \text{ ns}$$

$$T_{BC} = t_{CK-Q}(B) + t_{PD}(cc) + t_{SU}(C) = \\ 10 + 4 + 2 = 16 \text{ ns}$$

$cc$  is combinational  
clk.

$$\text{max clk freq. (} \Rightarrow T_{min} = 16 \text{ ns}) = \frac{1}{16 \text{ ns}} = \\ 62.5 \text{ MHz}$$

↳ Why ~~not~~ (A → B → cc → c) ?

↳ Validate ff hold time ( $t_h$ )  
hold time requirement on ffB → that  
the input.

to B should not change  
at least 2 ns after the rising  
edge of the clk.

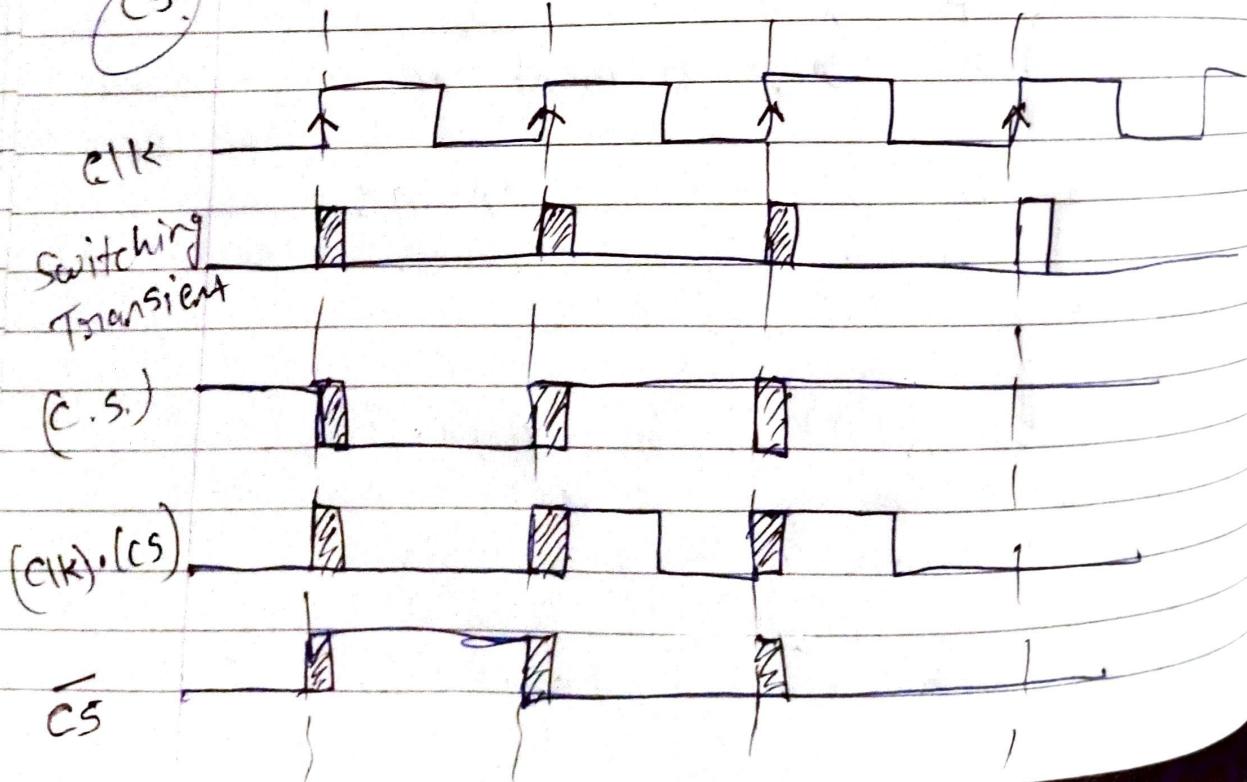
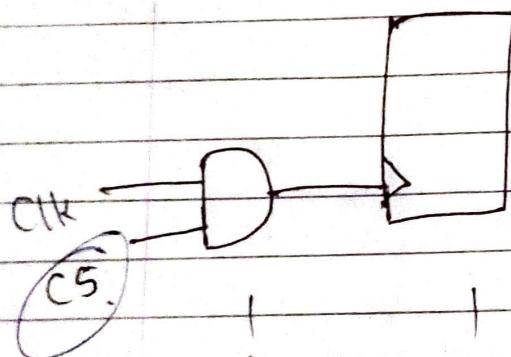
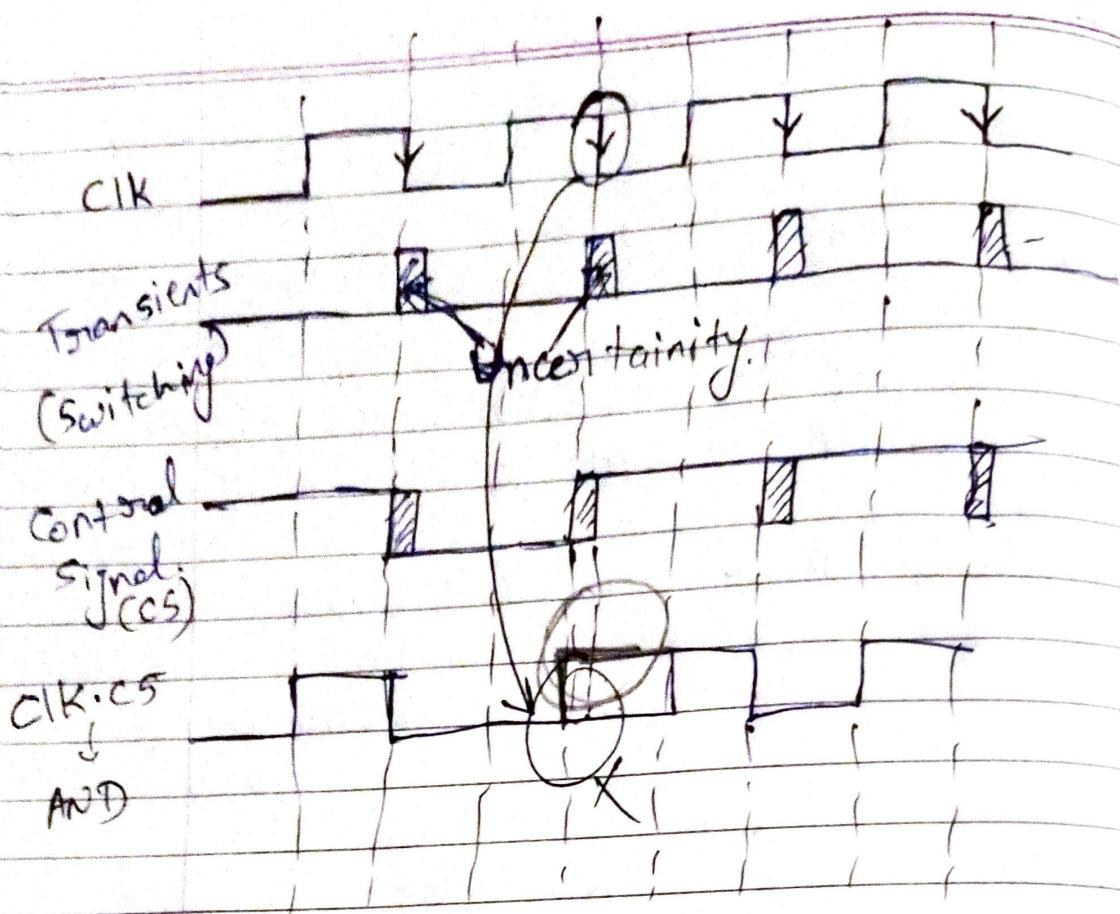
$$\hookrightarrow t_h(B) \leq t_{CD}(A) + t_{CD}(\text{comb})$$

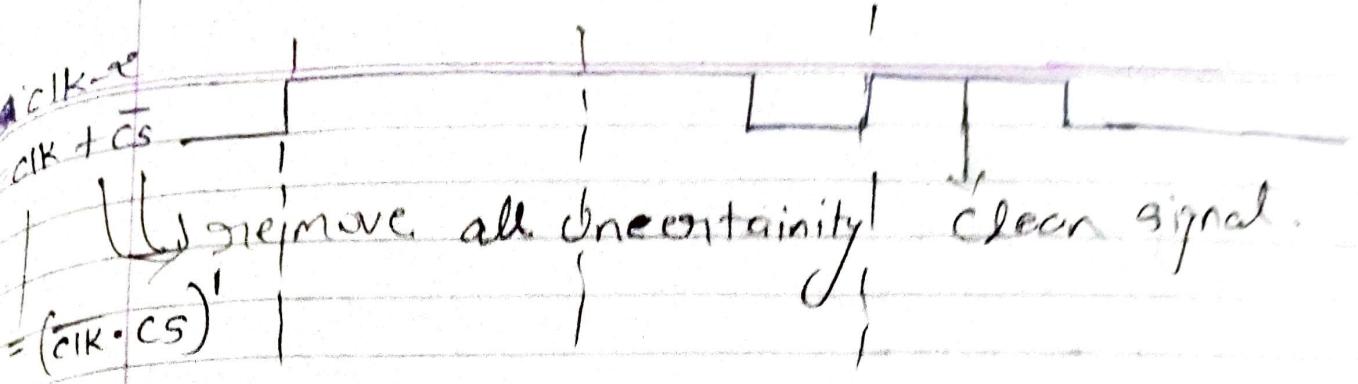
Since  $t_h$  (2 ns) is less than  $t_{CD}(A)$  (2 ns + 2 ns)

$$(1) t_h(B) \leq t_{CD}(A) \quad [\text{for } A \rightarrow B \text{ path}]$$

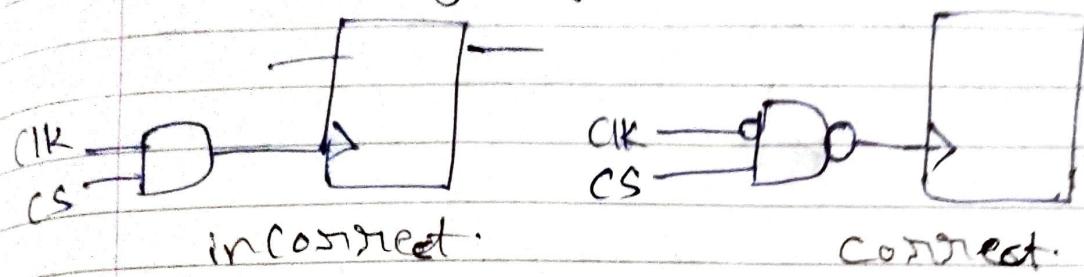
$$(2) t_h(C) \leq t_{CD}(A) + t_{CD}(cc) \quad [A \rightarrow C]$$

$$(3) t_h(C) \leq t_{CD}(B) + t_{CD}(cc) \quad [B \rightarrow C]$$



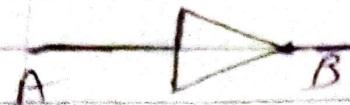


for rising edge



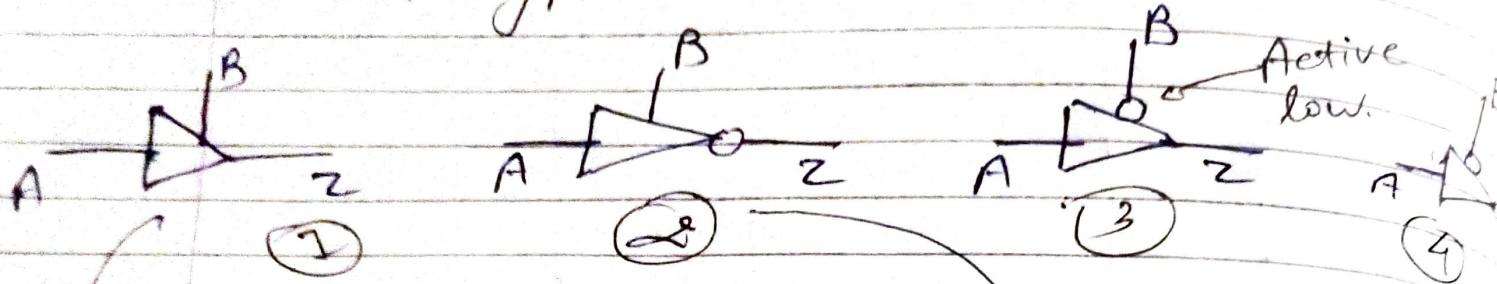
## \* Principles [Synchronous Design]

- 1] All clock i/p to the ffs, reg's, are by clk or a
- 2] All state changes occur immediately following the edge of the clk  
[setup, hold time has to be taken care of]
- 3] Advantage of such a design style (Synchronous) all transient, noise and other inconsistency of data that occurs between the clk will have no effect on the system performance and functionality.



Tristate drivers

↳ 4 types of



$\Delta \Rightarrow$  buffer =

(1)

B	A	Z
0	0	th-Z
0	1	th-Z
1	0	0
1	1	1

B	A	Z
0	0	0
0	1	1
1	0	1
1	1	0

$th-Z = \text{high } Z$

(3)

(4)

B	A	Z
0	0	0
0	1	1
1	0	th-Z
1	1	th-Z

B	A	Z
0	0	1
0	1	0
1	0	0
1	1	th-Z

