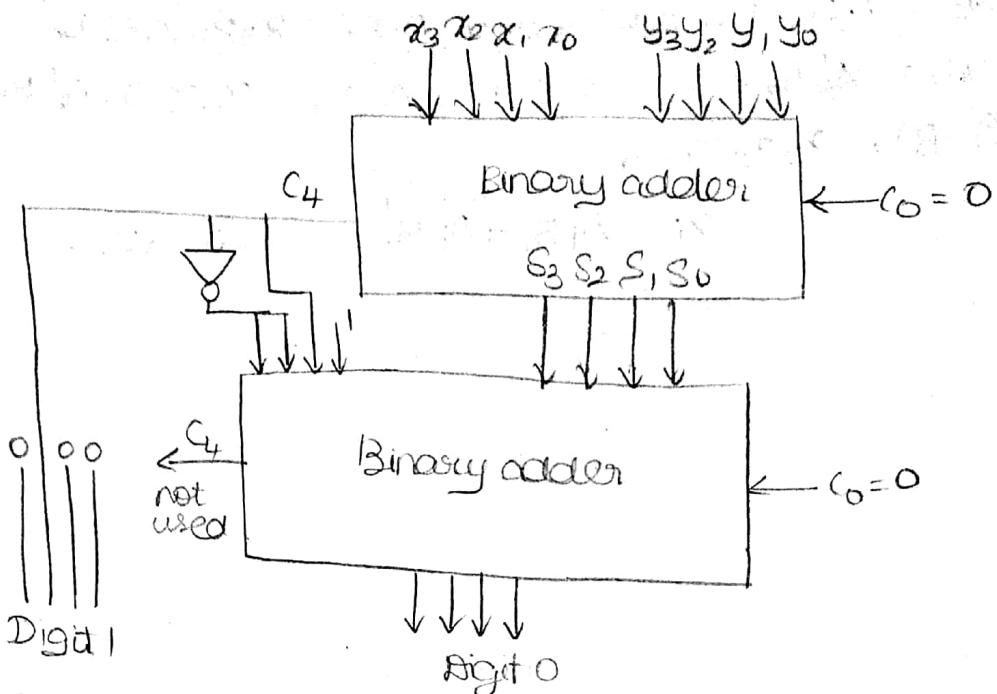


Excess 3 adder:



Priority encoder: The type of encoder available in IC form is called priority encoder.

Decimal to binary encoder:

10 Inputs \rightarrow 4 outputs

General purpose Processor
(Pentium)

CISC }

RISC \rightarrow Reduced Instruction Set Computer

immediate \searrow Instruction size \rightarrow same

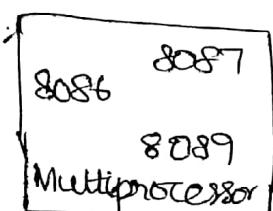
\hookrightarrow MVI 1,08 H \searrow No. of addressing modes \rightarrow less.

\downarrow (2 byte) \searrow MOV A,B \rightarrow moves the content
moves 08 into acc \searrow of B into acc.

STA 8000H

register. (1 byte).

\hookrightarrow LDA 8000H \rightarrow content of location 8000 is
Direct moved acc. (acc \rightarrow accumulator)
(3 byte)



System \rightarrow Address bus

bus \rightarrow Data bus

Microprocessor 886

(Default)

Bus arbitration:-

A technique used to share common resources.

8289 → Bus Arbitration

8288 → Bus controller

Application of encoder:

the code which is not processed by the processor will be converted to another form

Interrupt

↳ Divide by zero \rightarrow highest priority.

→ Power failure

Priority encoder:

small size \rightarrow 4 input 2+1 output
Inputs Output. \hookrightarrow valid bit.

	D_0	D_1	D_2	D_3	
	0	0	0	0	
1	0	0	0	0	
X	1	0	0	0	
X	X	1	0	0	
X	X	X	1	0	

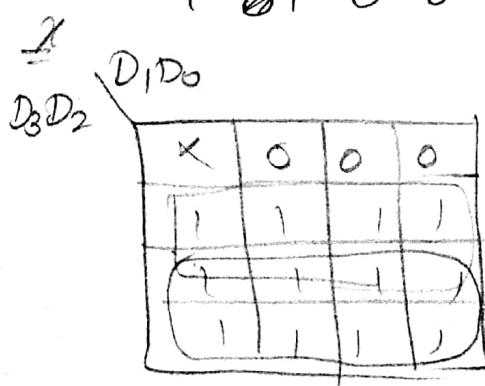
	x_1	x_2	v
	X	X	0
	0	0	1
	0	1	1
	1	0	1
	1	1	1

~~Kmap for z_c .~~

~~Knap for y~~

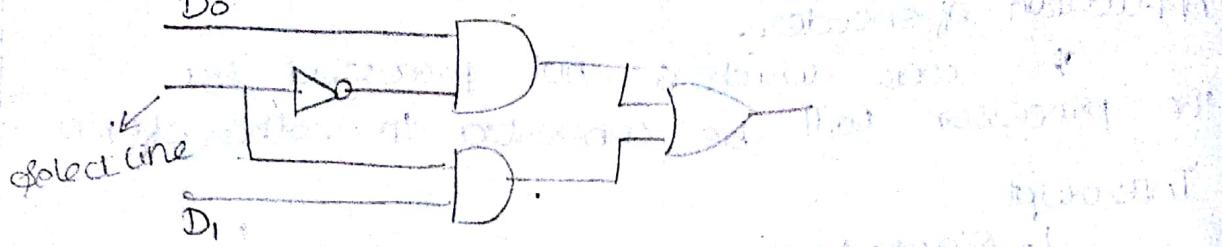
D_1	D_2	$D_1 D_2$	$D_1 \oplus D_2$	$D_1 \otimes D_2$
0	0	0	01	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
1	0	0	1	0
1	0	0	1	1
1	0	0	0	0
1	0	0	1	1
1	0	0	1	0
1	0	0	1	1
1	1	0	0	0

x	y	v
x	x	v
o	o	
o	1	1
o	1	1
1	0	1
1	0	1
1	0	1
1	0	1
1	1	1
1	1	1
1	1	1
1	1	1



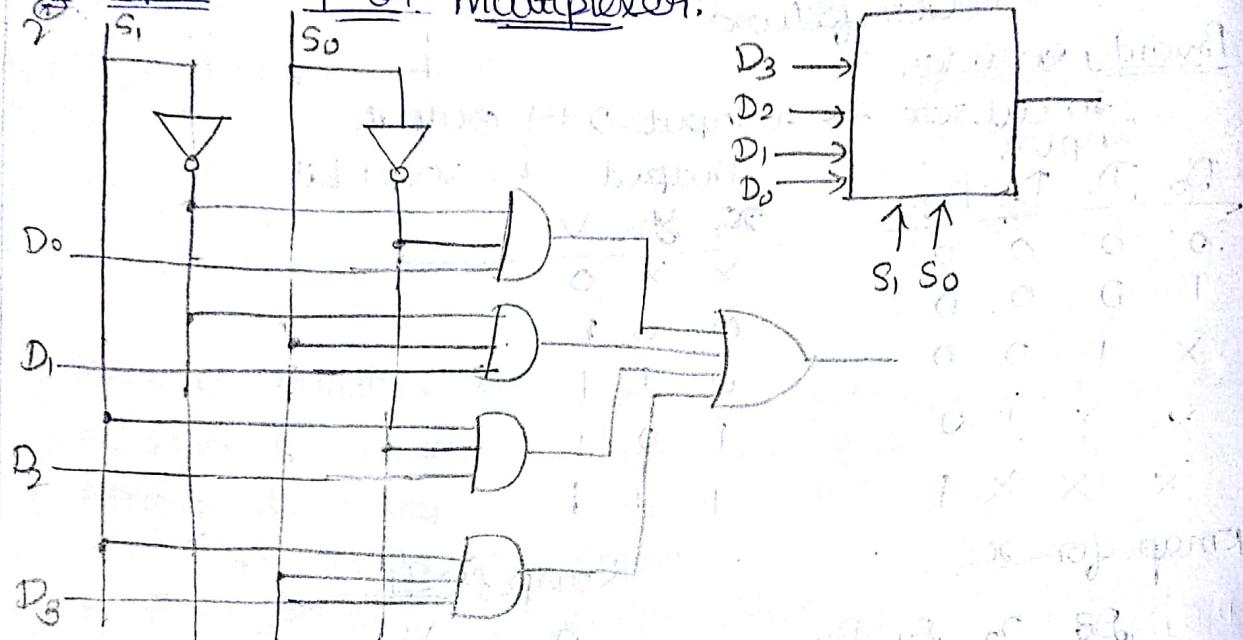
Multiplexers: \rightarrow n number of inputs & 1 output

2 inputs: NO. of select lines =

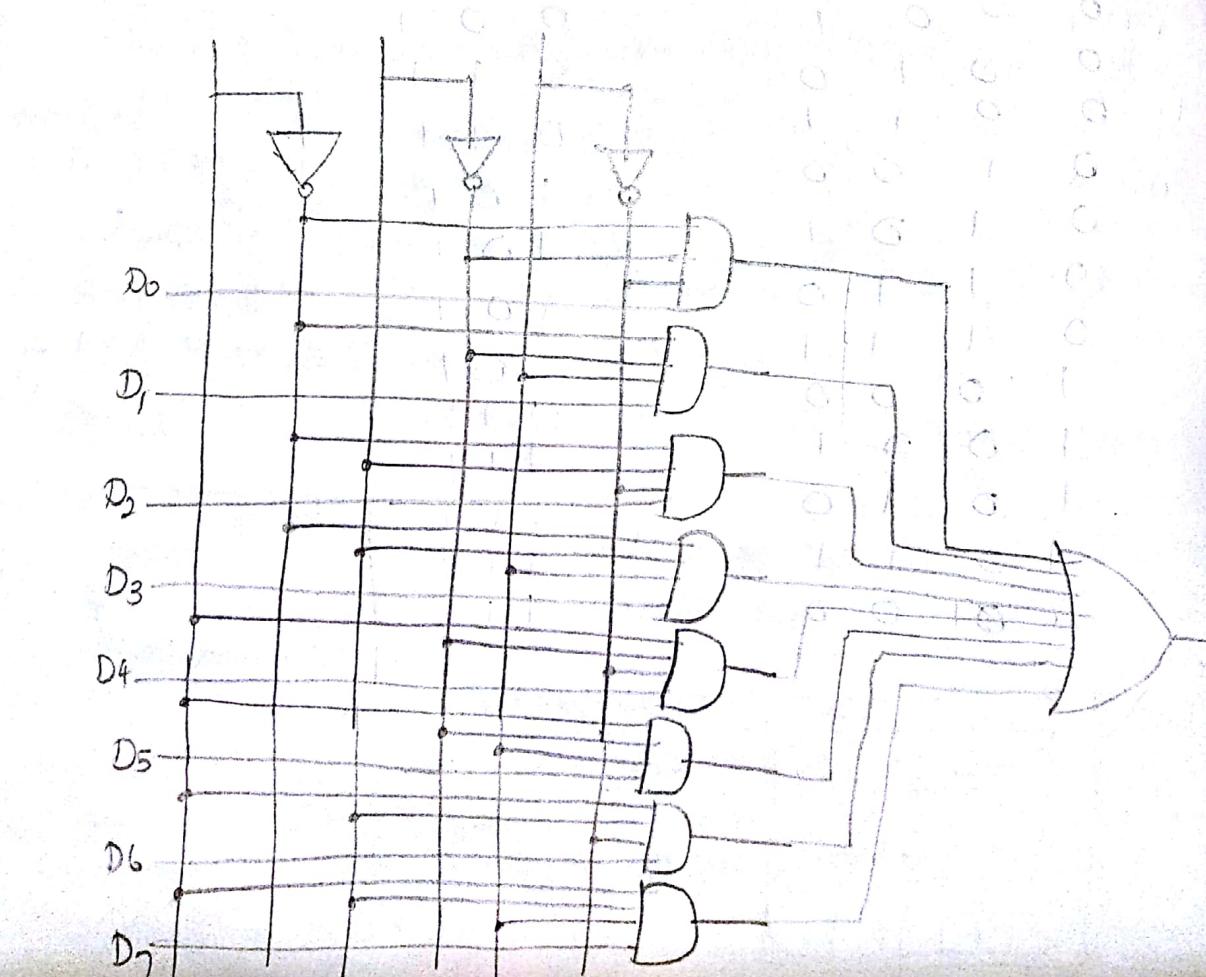


4 inputs:

4 to 1 multiplexer:



8 to 1 multiplexer:



$$F(xyz) = \Sigma_m(1, 2, 6, 7)$$

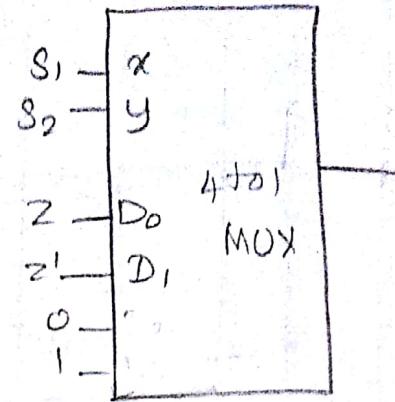
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$F=z$$

$$F=z'$$

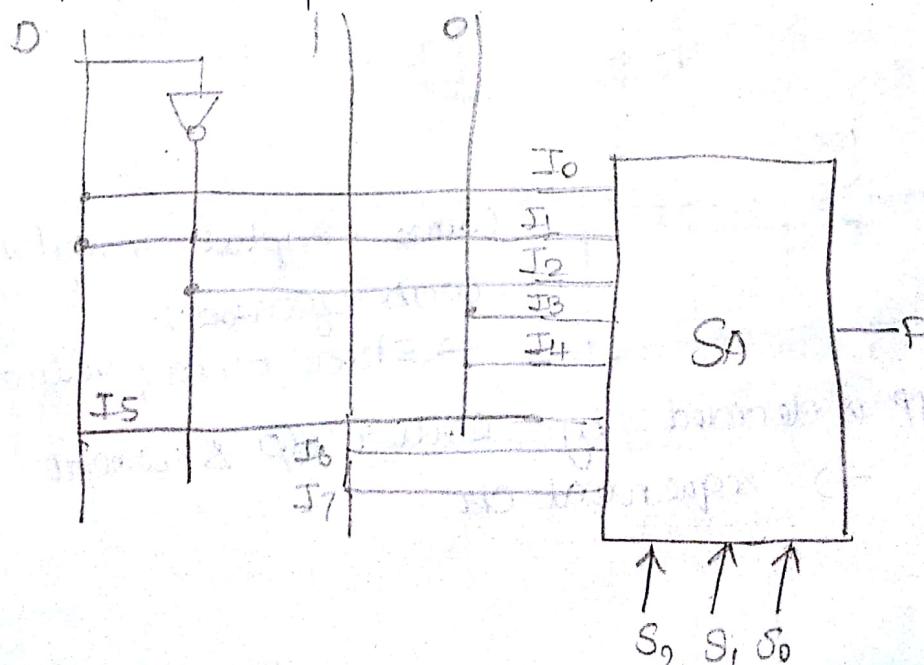
$$\Rightarrow F=0$$

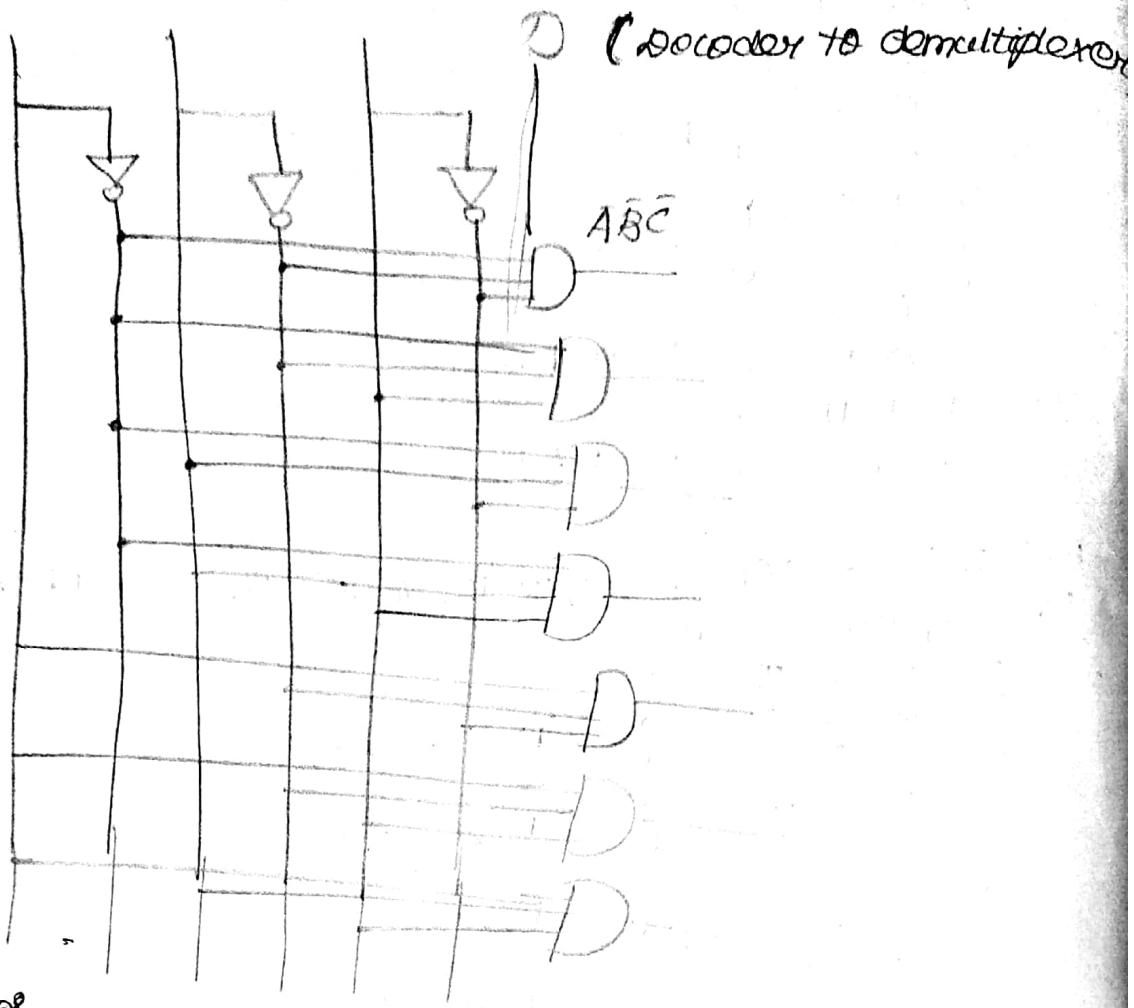
$$\Rightarrow F=1$$



$F(A, B, C, D) = \Sigma(1, 3, 4, 11, 12, 13, 14, 15)$ Use 8-to-1 MUX
to implement it.

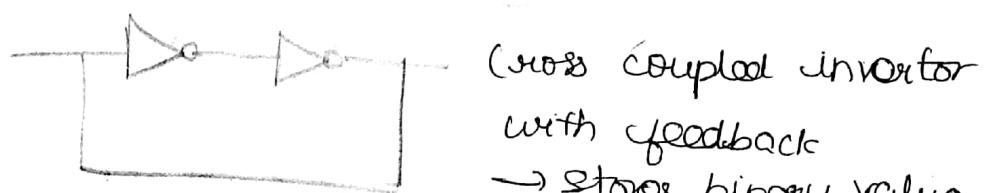
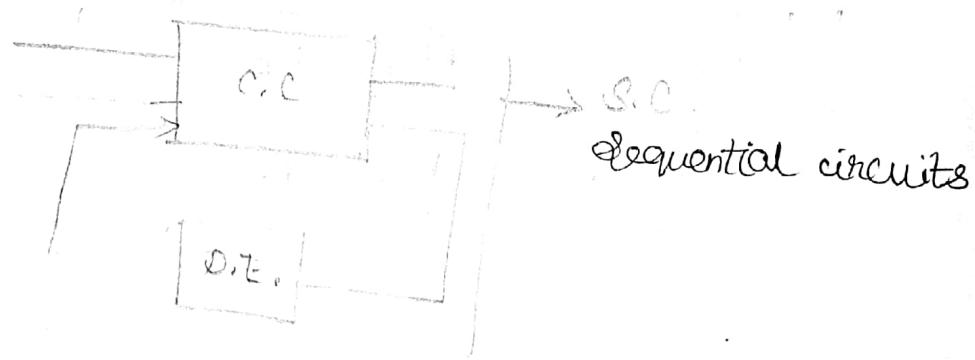
A	B	C	D	F	
0	0	0	0	0	$F=z$ D ₀
0	0	0	1	1	
0	0	1	0	0	$F=z$ D ₁
0	0	1	1	1	
0	1	0	0	1	$F=z'$ D ₂
0	1	0	1	0	
0	1	1	0	0	$F=0$ D ₃
0	1	1	1	0	
1	0	0	0	0	$F=0$ D ₄
1	0	0	1	0	
1	0	1	0	0	$F=z$ D ₅
1	0	1	1	1	
1	1	0	0	1	$F=1$ D ₆
1	0	0	1	1	
1	1	1	0	1	$F=1$ D ₇
1	1	1	1	1	





Sequential Circuits.

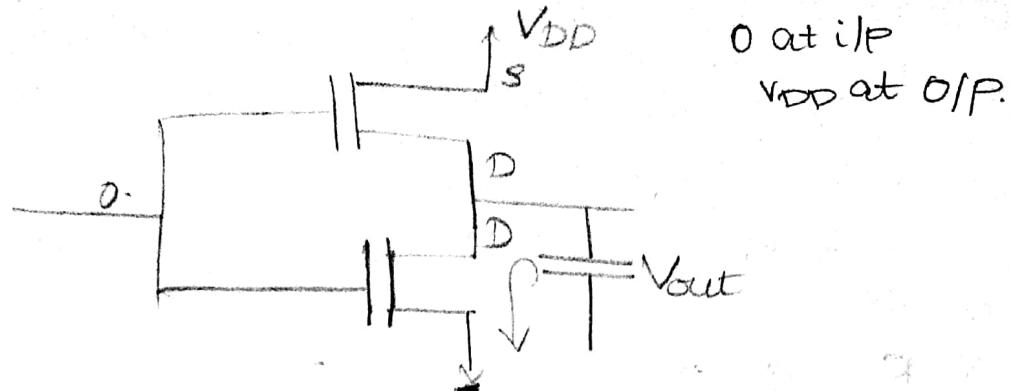
The hardware used to perform arithmetic / logical operations are called data path.



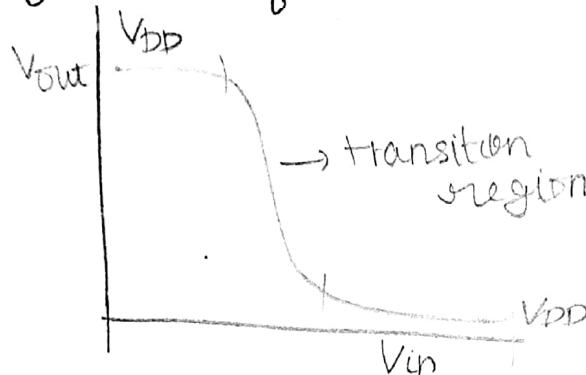
Cross coupled inverter
with feedback
→ stores binary value

O/P is decided by earlier o/p & current i/p \Rightarrow sequential circuit.

(ver)



Voltage transfer characteristics



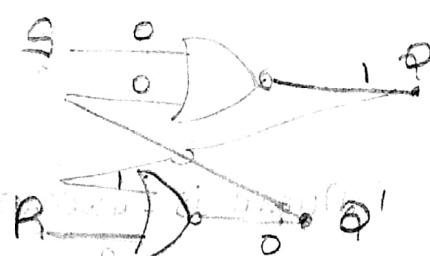
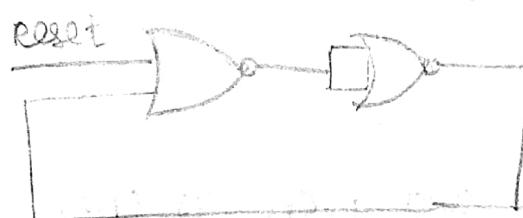
To avoid power dissipation we need sharp transition.



In inverter transition

Happens at $V_{DD}/2$

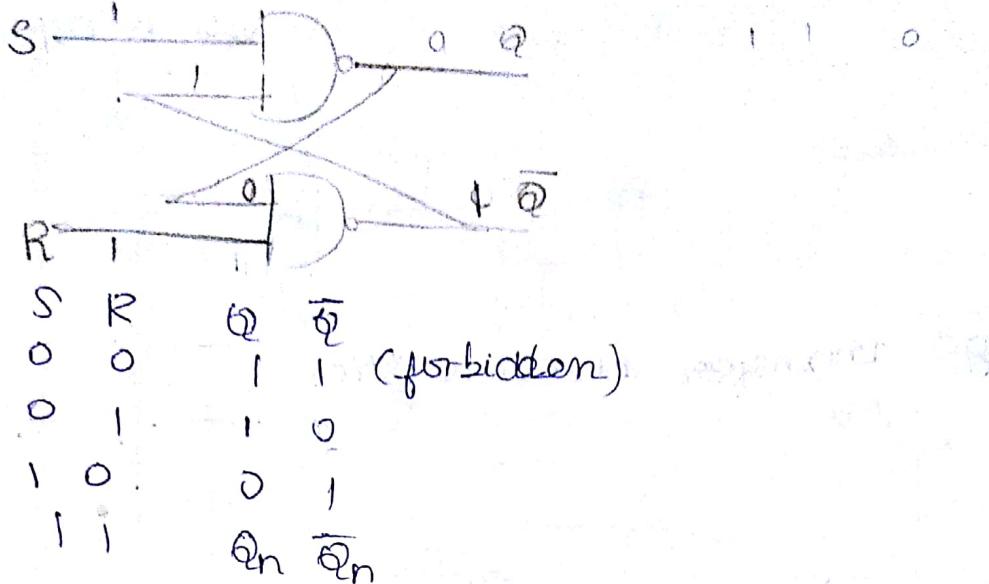
Realisation of inverter by NOT gate:



S	R	Q	\bar{Q}
0	0	Q_n	\bar{Q}_n
0	1	0	1
1	0	1	0
1	1	0	0

bcz Q and \bar{Q} are same

(forbidden state)



Sequential circuits.

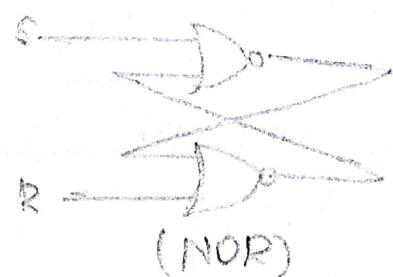
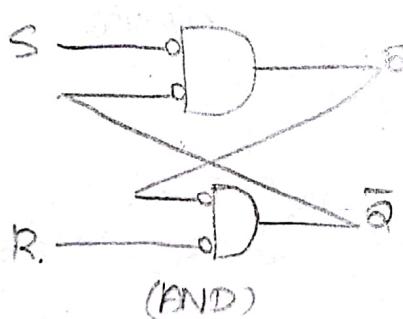
Synchronous:



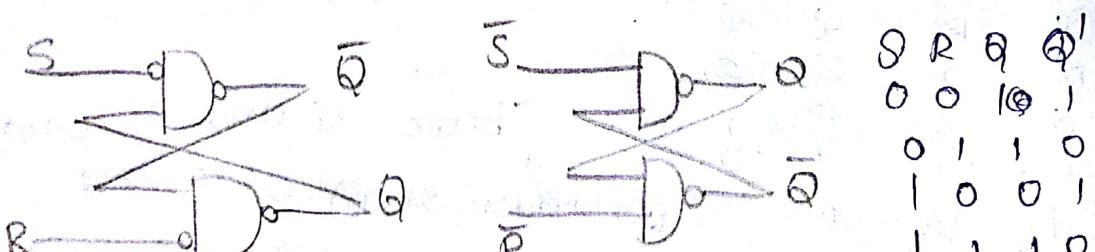
Edge triggered

Level triggered

SR flip flops converted to NAND gates



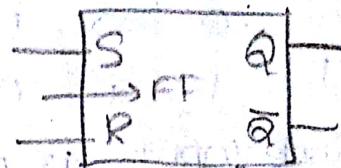
Output is inverted and given as Q/p:



Nand implementation

\bar{S}	\bar{R}	Q_n	Q_{n+1}
0	0	0	0
0	1	0	1
0	0	1	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

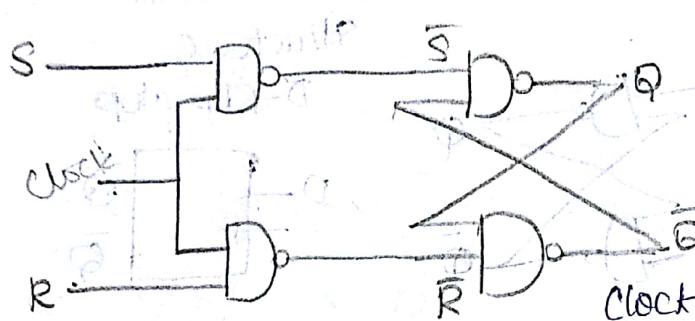
Symbol of SR flip-flops



clock i/p: /sometimes realised as

enable i/p:

If don't give the clock it retains the previous stage



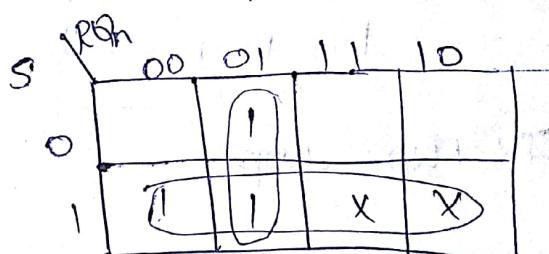
Clock will be

S	R	Q_n	Q_{n+1}
0	0	0	Next stage continuously running, but enable / disable is depend
0	1	0	Current stage upon our decision.
1	0	1	Reset
1	1	?	Set

Function table

\bar{S}	\bar{R}	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	?
1	1	1	?

K-map for SR flip-flops



$S + \bar{R} Q_n$

\Rightarrow forbidden state

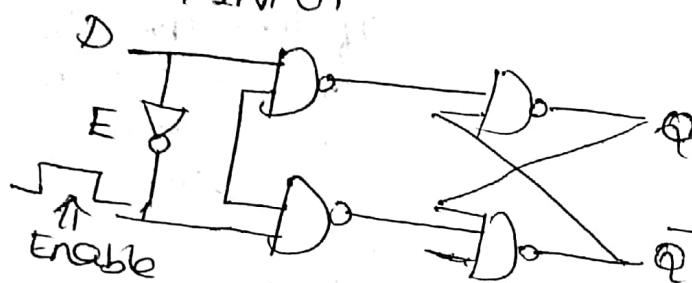
Excitation method:

Excitation Table:

Q_n	Q_{n+1}	S	R
0	0	0	0
0	1	1	0
1	0	0	1
1	1	x	0

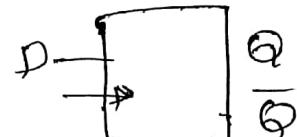
* clock input is used in PLL, clock operate at high speed, we must decide its charging & discharging.

DELAY INPUT



Structure

D-Flip flop



Functional Table:

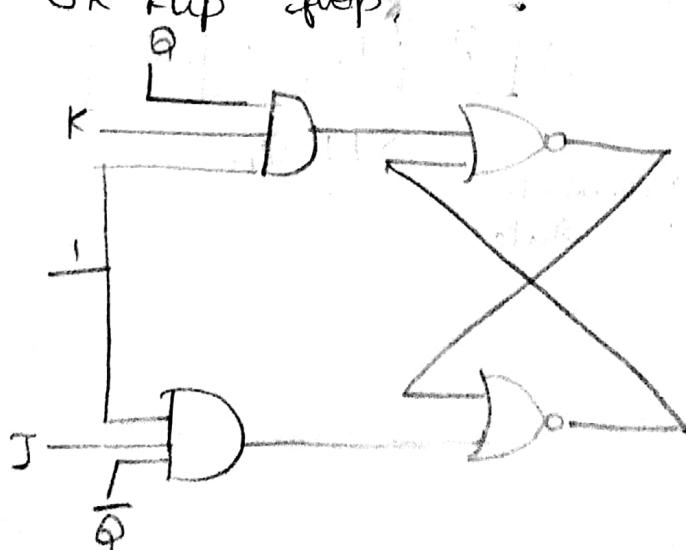
E_n	D	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

E_n	DQ_n	00	01	11	10
00	00	1	1	1	1
01	01	1	1	1	1
11	11	1	1	1	1
10	10	1	1	1	1

Excitation Table

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

JK Flip flop:



J	K	Q_n	Q_{n+1}
0	0	0	$Q_n = 0$
0	0	1	$Q_n = 1$
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

JK

Q_n	00	01	11	10
0	0	1	0	1
1	1	0	1	0
	D	1	1	C

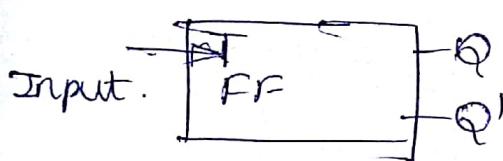
$$J\bar{K}Q_n + JK\bar{Q}_n + \bar{E}Q_n$$

$$J\bar{Q}_n + \bar{E}Q_n$$

Excitation Table:

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

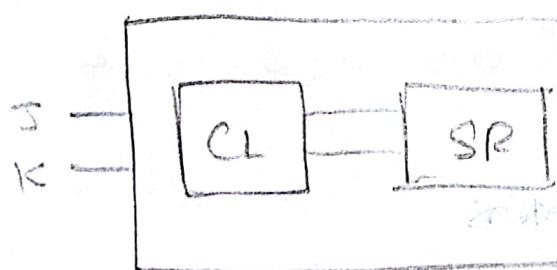
T-flip flop:



T	Q_{n+1}
0	Q_n
1	\bar{Q}_n

T-flip flops can be used to divide frequency by 2.

SR flip flop can be used to realise JK flip flop

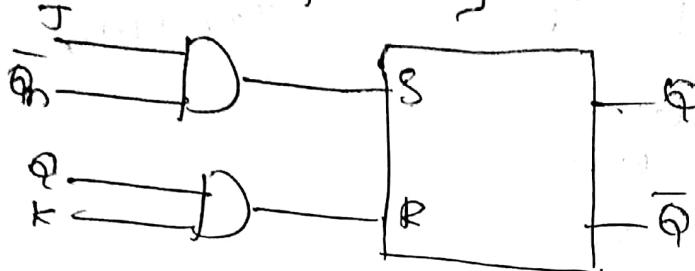


J	K	Q_n	Q_{n+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

J	KQn	Q	Q'
0	0	0	1
1	0	1	0
0	1	X	X
1	1	1	0

don't care
 $S = KQn + JQ'n$
 $S = JQ'n$

JK flip flop using SR:



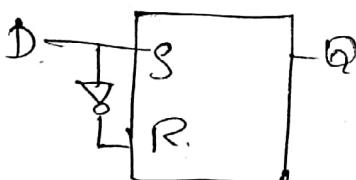
D flip flop using SR flip flop:

D Q_n Q_{n+1} S R.

0	0	0	0	x
0	1	0	0	1
1	0	1	1	0
1	1	1	x	0

D	Q_n	Q_{n+1}
0	0	0
1	1	x

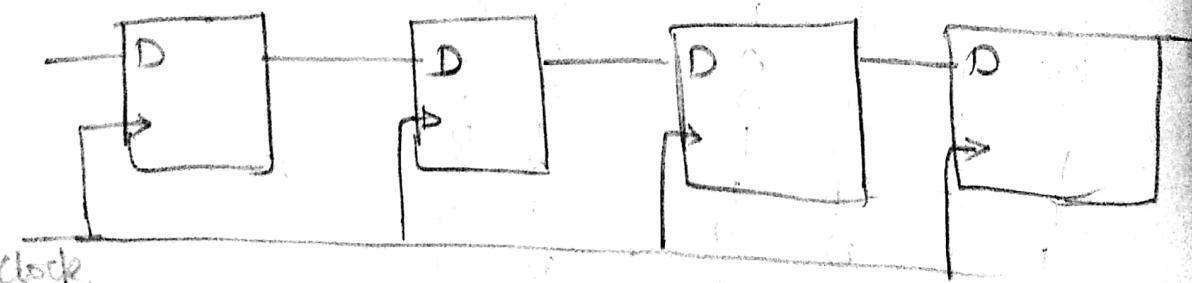
D	Q_n	R
0	x	1



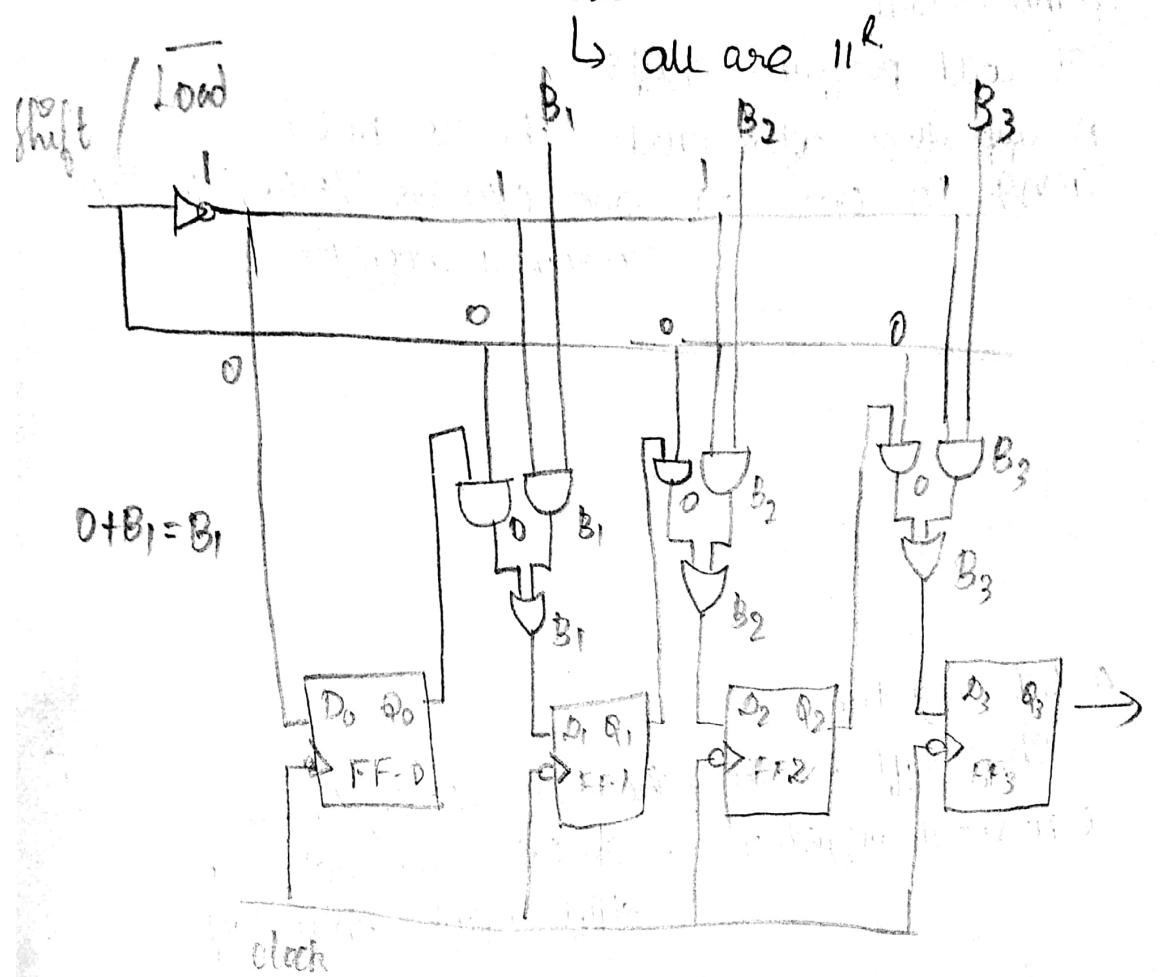
Registers.

→ Storage devices

→ Combination of many flip flop.



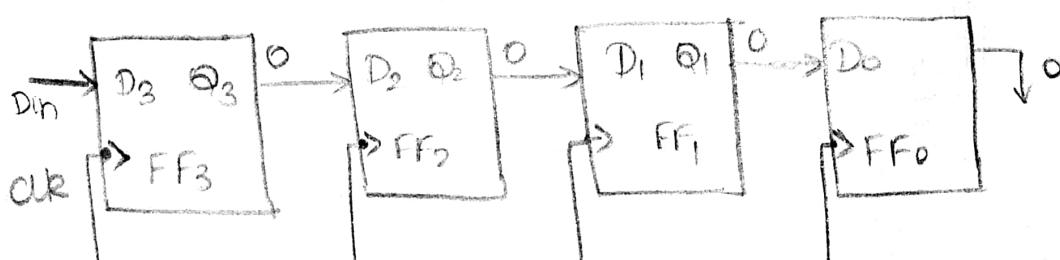
- 1) Serial in / Serial out \rightarrow i/p given immediate o/p
 2) Serial in / Parallel out \rightarrow i/p given \Rightarrow all o/p are seen at one time
 3) Parallel in / Serial out \rightarrow all i/p given \Rightarrow o/p one by one
 4) Parallel in / Parallel out



Q) Load Mode

2) Shift Mode

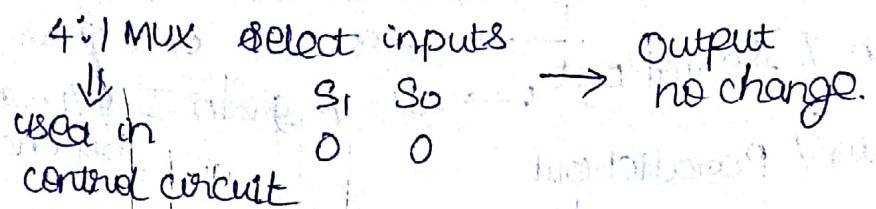
Shift Right mode:



Clk	Q3	Q2	Q1	Q0
↓	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1

Clk	D	Yout
0	X	Qn
1	0	0
1	1	1

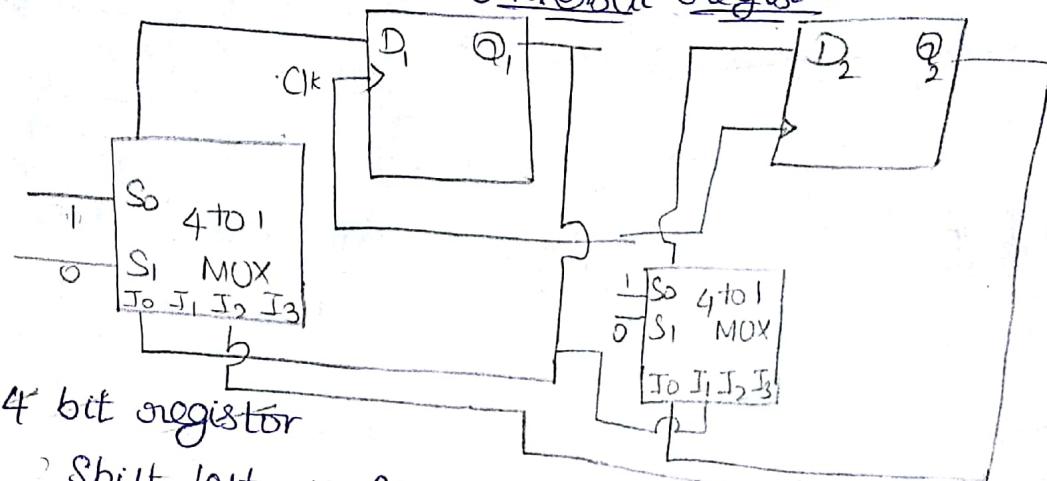
SHL - Shift Left Asynchronous
 SHR - Shift Right. → preset
 ↘ clear / Reset



It will perform shifts.

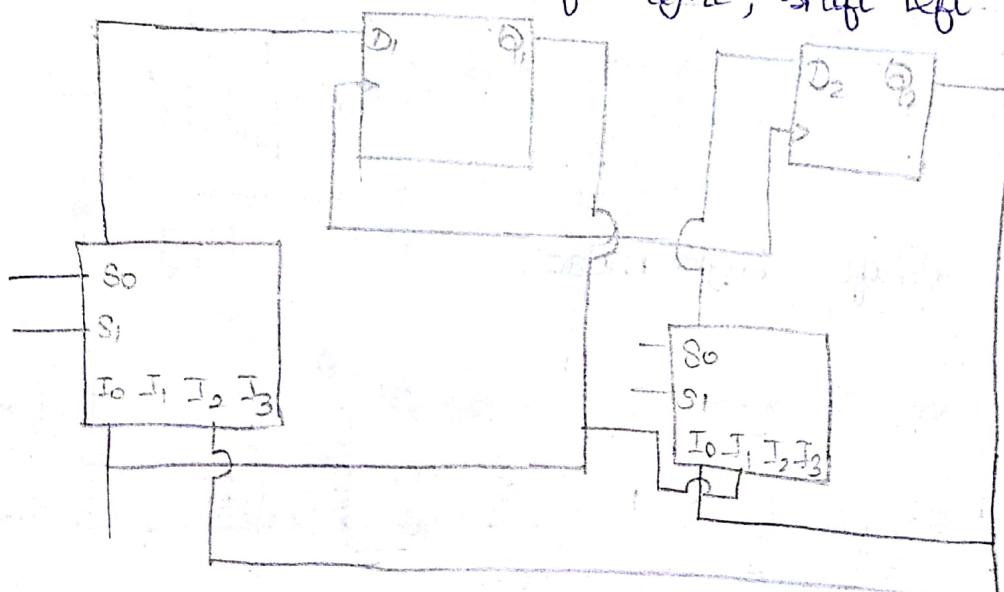
A flip flop after clock should not have
 change in O/P \Rightarrow give O/P as i/p to flip flop

Universal register:



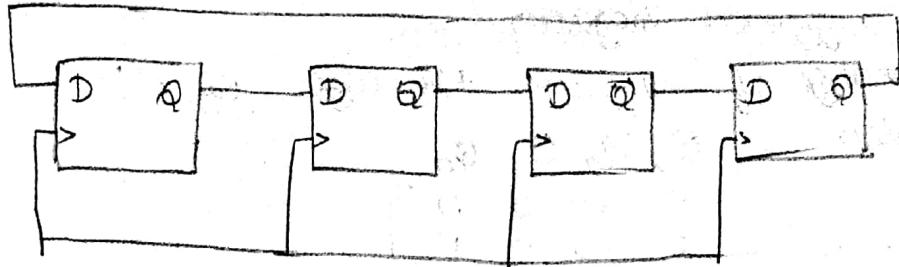
Shift left \Rightarrow Give the O/P at Q_2 to I_2

Universal register \Rightarrow serial, parallel,
 shift right, shift left.



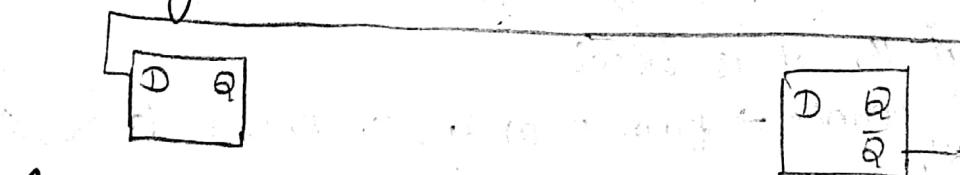
Ring Counter:

Output of last stage is connected to
 the input of first FF



Johnson Counter:

Q of last stage is connected as the D i/p of 1st stage.



Counter \rightarrow used to count no. of clock pulses.

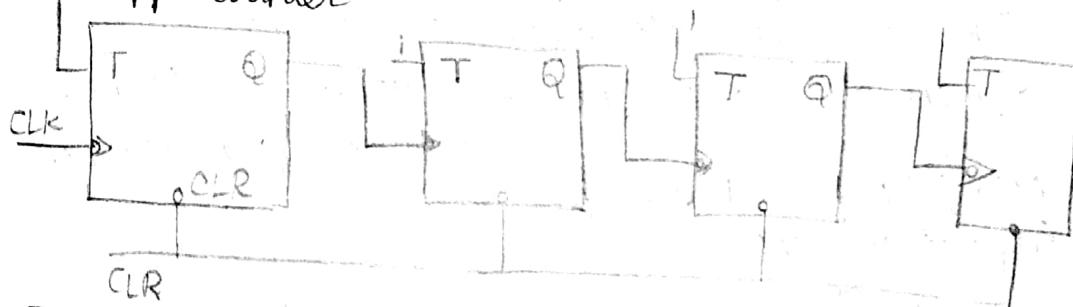
sipple
(or) Asynchronous \downarrow \rightarrow synchronous

$mod-N \Rightarrow$ goes through n no. of stages before it is reset.

Clock is given to 1st stage of counter \Rightarrow Asyn.

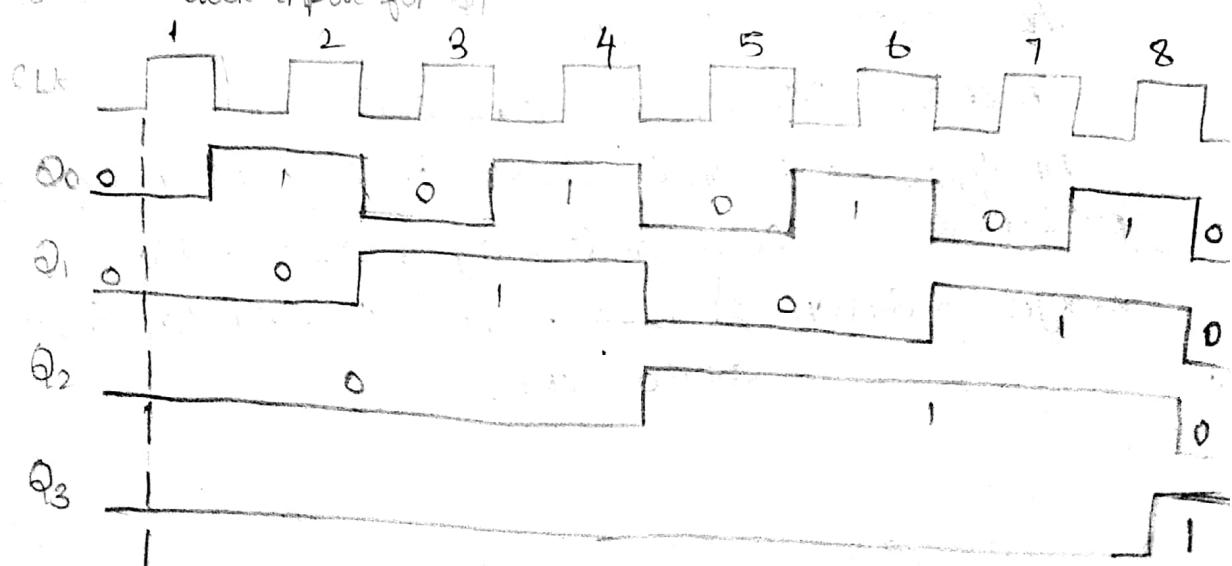
Clock is given to all stages of counter \Rightarrow syn.

|| Ripple counter



If CLR i/p is 0 \Rightarrow O/p is 0

CLR i/p is 1 \Rightarrow the count operation will occur
@ the clock input for Q₁



the edge triggered register

↳ change of o/p in the edge of clk

clk pulse	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1
5	1	1	1	0
6	1	1	0	0
7	1	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0

mode 16

From 16 it is reset

Decade counter \Rightarrow from 0 to 9, 10 onwards reset

mode 10 counter

state diagram

state table

The delay of each flip flop is not same.

FSM \Rightarrow Finite State Machine \Rightarrow Synchronous sequential ckt

Synch counters \Rightarrow The previous output is the clock of next stage.

Synch \Rightarrow same clock for all stages.

Synch ckt \rightarrow 2 types \rightarrow Moore ckt / Machine

\rightarrow Mealy ckt / machine

Delay element \rightarrow Flip flop.

Sequential ckt

If the present state of ckt decides the o/p \Rightarrow Moore

If the state and input of ckt decides the o/p \Rightarrow Mealy

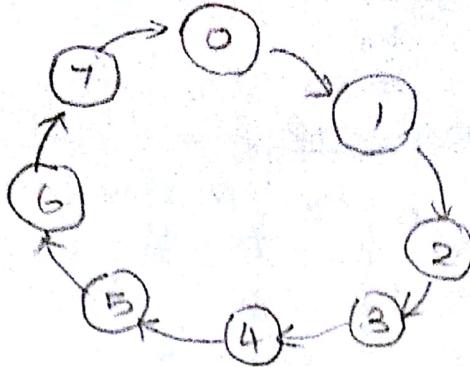
In short Moore \Rightarrow present state

Analysis \Rightarrow Given logic ckt write boolean expression procedure.

find state diag, state table

Design \Rightarrow Given prob \Rightarrow state diag, state table
draw ckt.





Each state replaced by equivalent binary no.
from the excitation table.

Present state	Next state	J_2	K_2	J_1	K_1	J_0	K_0
$Q_2\ Q_1\ Q_0$	$Q_2\ Q_1\ Q_0$						
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	x	0	0	x	1	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 0 0	x	1	x	1	x	1

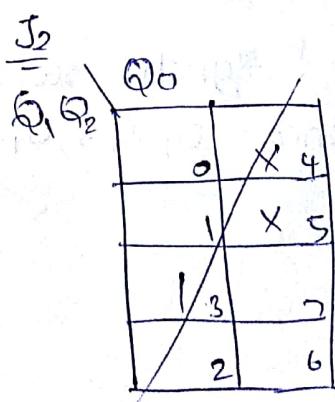
Excitation table of JK

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

x denotes don't care. (i.e) either 0 or 1.

Use K-map to find out $J_2\ K_2\ J_1\ K_1\ J_0\ K_0$
considering present state.

$J_2\ Q_1\ Q_0$					
Q_2	Q_1	00	01	11	10
0	0	0	1	1	0
1	x	4	x	5	x



		J_2	Q_0	Q_1	Q_2
		0	1		
		00	0	1	
01		2	1	3	
11		X ₆	X ₇		
10		X ₄	X ₅		

		K_2	Q_1	Q_0		
		0	00	01	11	10
		1	X	X	(X)	X

$$k_2 = Q_1 \oplus Q_0.$$

		J_1	Q_2	Q_1	Q_0
		00	01	11	10
		0	(1)	X	X
1		X	(1)	X	X

		K_1	Q_2	Q_1	Q_0
		00	01	11	10
		0	X	X	1
1		X	X	1	

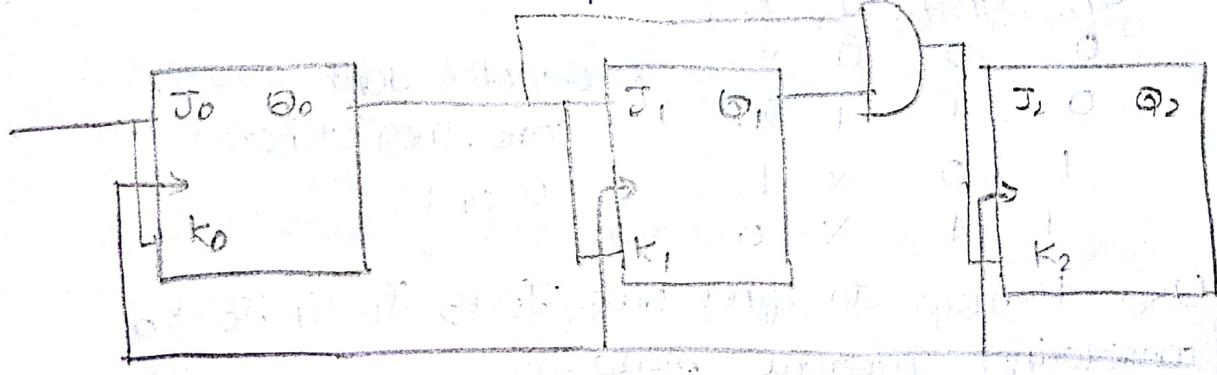
$$k_1 = Q_0.$$

		J_0	Q_2	Q_1	Q_0
		00	01	11	10
		0	1	X ₁	X ₂
1		1	X ₄	X ₅	1

$$= Q_0$$

		K_0	Q_2	Q_1	Q_0
		00	01	11	10
		0	X	1	1
1		X	1	1	X

$$= Q_0.$$



Design a synch seq. clk. counter to count the sequence 0, 2, 3, 6, 7, 0.

Present state	Next state	
$Q_2\ Q_1\ Q_0$	$Q_2\ Q_1\ Q_0$	$D_2\ D_1\ D_0$
0 0 0	0 1 0	0 1 0
0 0 1	- - -	x x x
0 1 0	0 1 1	0 1 1
0 1 1	1 1 0	1 1 0
1 0 0	- - -	x x x
1 0 1	- - -	x x x
1 1 0	1 1 1	1 1 1
1 1 1	0 0 0	0 0 0

Excitation
Table of
D flipflop

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

$\underline{D_2}$	$Q_1\ Q_0$	00	01	11	10
Q_2	0	(x)	1		
	1	x	x		1

$$= \bar{Q}_2 Q_0 + \bar{Q}_0 Q_2$$

$$D_2 = Q_2 \oplus Q_0$$

$\underline{D_0}$	$Q_1\ Q_0$	00	01	11	10
Q_2	0	x			1
	1	x	x		1

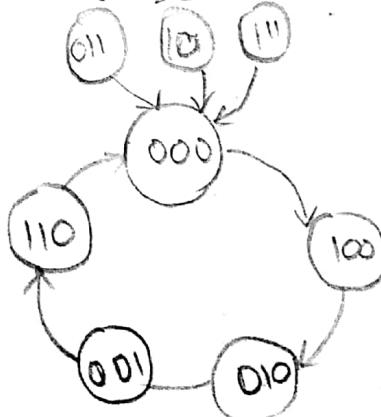
$$D_0 = Q_1 \bar{Q}_0$$

0, 4, 2, 1, b

State diag.

Unused states

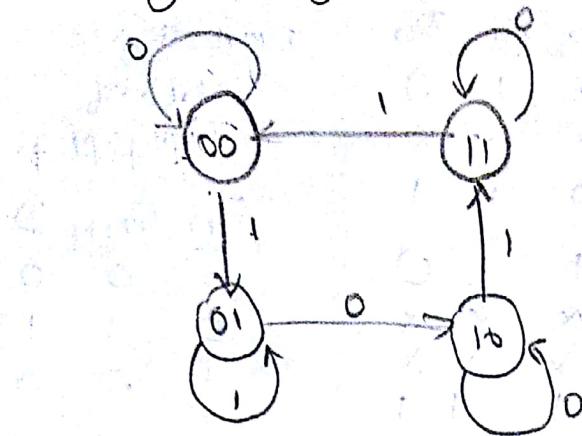
011, 101, 111



The T flip flop

Lock out state
⇒ cycling thru
unused states

Design a synchronous seq. ckt whose state diag is given



state table:

Present state			Next stage	
Q_1	Q_0	X	Q_1	Q_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Using JK flip flop
Excitation Table

$Q_n \ Q_{n+1} \ J \ K$

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

J_1	$Q_0 X$	00	01	11	10
Q_1	0				1
1	x	x	x	x	x

$$J_1 = Q_0 \bar{X}$$

J_0	$Q_1 X$	00	01	11	10
Q_1	0	x	x	1	1
1	x	x	x	x	x

$$K_0 = Q_1 \bar{X} + \bar{Q}_1 X$$

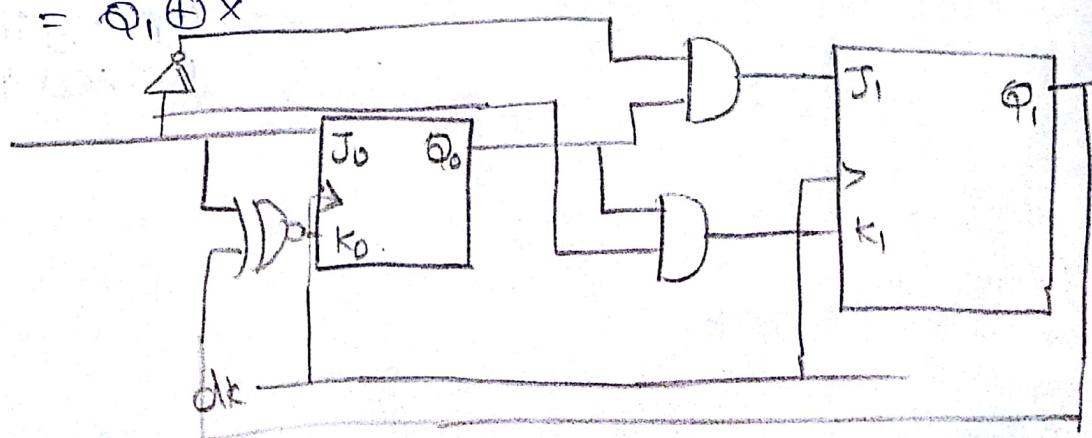
$$= Q_1 \oplus X$$

K_1	$Q_0 X$	00	01	11	10
Q_1	0	x	x	x	x
1	x	x	x	x	x

$$= \bar{X} + \bar{Q}_0$$

J_0	$Q_1 X$	00	01	11	10
Q_1	0	1	x	x	x
1	x	x	x	x	x

$$J_0 = X$$



$J_1 \ K_1 \ J_0 \ K_0$

0 X 0 X

0 X 1 X

1 X X 0

0 X X 1

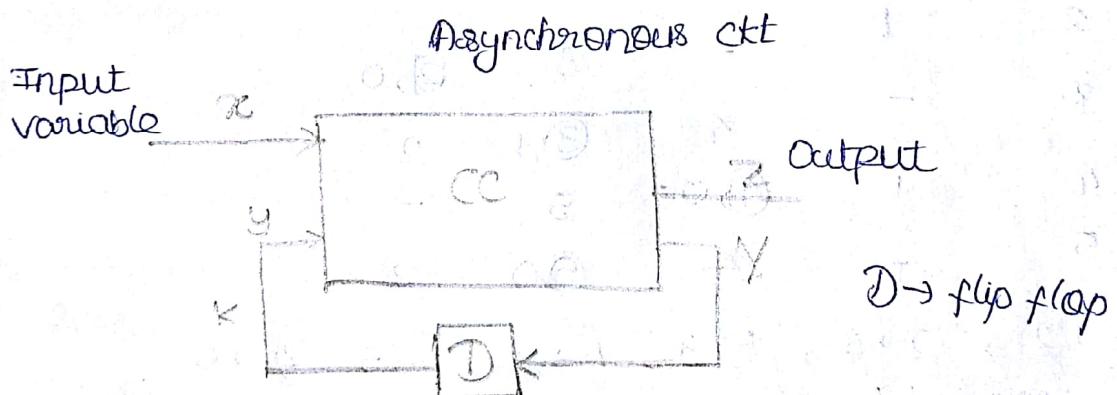
X 1 0 X

X 1 1 X

X 1 X 1

X 0 X 0

Design a seq. Ckt with 2 JK flip flop and 2 IP E and X. If $E=0$ the ckt remains in same state regardless of value of X . $E=1, X=1$ the ckt goes state transition from 00 to 01 to 10 to 11 and to 00 & repeats. $E=1, X=0 \Rightarrow 00$ to 11 to 10 to 01 and back to 00 & repeats.

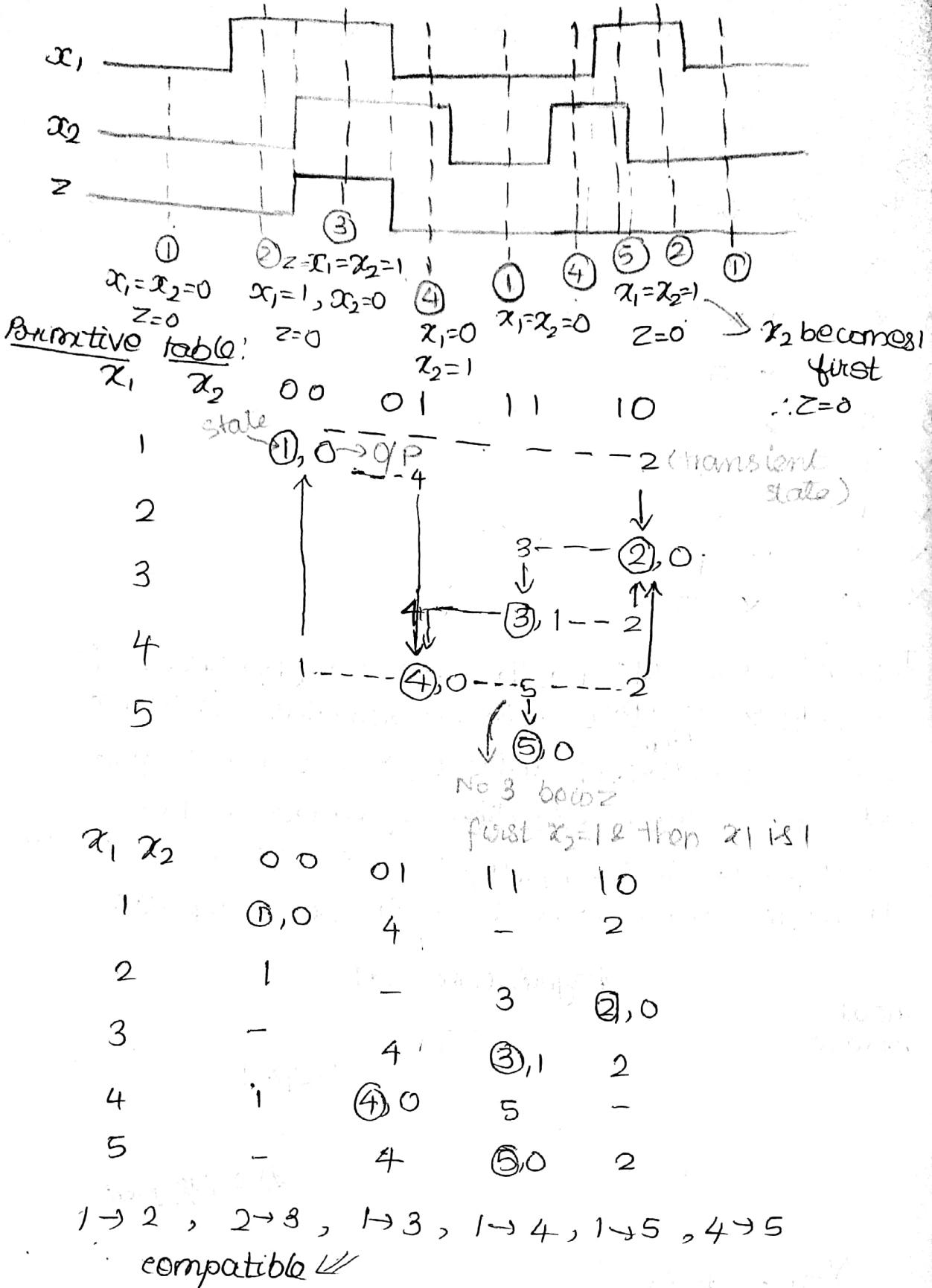


$$Y=y \Rightarrow \text{stable}$$

Fundamental mode

↳ 1 input variable to change its state before the Ckt gets into stable state.

- Consider the ckt with 2 I/p x_1 & x_2 & 1 O/p z
The O/p is 1 when x_1 & x_2 are 1 with x_1 being 1 first



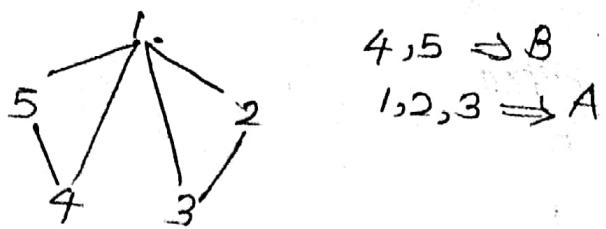
$1 \rightarrow 2, 2 \rightarrow 3, 1 \rightarrow 3, 1 \rightarrow 4, 1 \rightarrow 5, 4 \rightarrow 5$

compatible //

$2 \rightarrow 4, 3 \rightarrow 4, 2 \rightarrow 5, 3 \rightarrow 5 \Rightarrow$ NOT compatible

Group the compatible pairs:

clique \rightarrow subgraph where all the nodes are connected to each other



$4,5 \Rightarrow B$
 $1,2,3 \Rightarrow A$

$x_1 x_2$ 00 01 11 10

A 1,0 4,0 3,1 2,0

$x_1 x_2$ 00 01 11 10
B 1,0 4,0 5,0 2,0

A(0) 0,0 1,0 0,1 0,0 (y, z)

B(1) 0,0 1,0 1,0 0,0

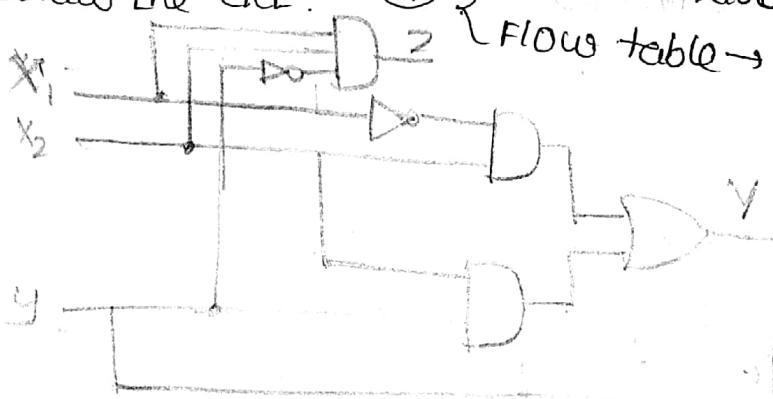
Kmap for Y:

		$x_1 x_2$	00	01	11	10
		y	0	1	0	0
x	0	0	1	0	0	0
		1	0	1	0	0

$$Y = yx_2 + \bar{x}_1 x_2$$

Draw the ckt.

④ Transition Table \rightarrow states in binary
Flow table \rightarrow states in alphabets.



- 2) Design a gate circuit when the input G-gate, D-gate
Consider g=1, Output=D, When Q₁=~~Q₂~~, O/P is obtained.

DG

Ⓐ

01/0

11/1 Ⓑ

Ⓒ

00/0 10/0

Ⓓ

10/1 Ⓟ

00/1 Ⓠ

Ⓕ

Inputs

RS	D	G	Output
A	0	1	0
B	1	1	1
C	0	0	0. After state A or D
D	1	0	0 After state C
E	1	0	1 After B or F
F	0	0	1 After E

Primitive table:

RS	Next State DG.				only one stable stable in a row but only 1 variable change.
	00	01	11	10	
(0,1) A	C,-	A,0	B,-	D,-	
(1,1) B	E,-	A,-	B,1	E,-	
(0,0) C	0,0	A,-	-,-	D,-	
(1,0) D	C,-	-,-	B,-	D,0	
(1,0) E	E,-	-,-	B,-	E,D	
(0,0) F	F,-	A,-	-,-	E,-	Change input by same O/P

Implication Table

	B	C	D	E
B	✓			
C	✓	DE		
D	✓	BE	✓	
E	CE	✓	CFR	X
F	CF	✓	X	CF

Column E : (E, F)

COL D : (E, F)

COL C : (E, F) (C, D)

COL B : (B, E, F) (C, D)

COL A : (B, E, F) (A, C, D)

compatible



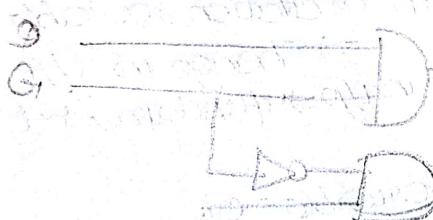
PS	NS			
	00	01	11	10
B	- - A, - (B), E, -			
E	F, - - B, - (E), Y			
F	(F), A, - - - E, -			

ACB	NS			
	00	01	11	10
C, 0	A, 0	B, -	D, 0	
F, 1	A, -	B, Y	E, Y	

y	NS			
	00	01	11	10
0	0, 0	0, 0	1, -	0, 0
1	1, 1	0, -	1, 1	1, 1

y	NS			
	00	01	11	10
0	0 0 X 0			
1	1 X 1 1			

$$Z = y$$



A	NS			
	00	01	11	10
C, -	0, 0	B, -	-	
C, 0	A, -	-	D, -	
C, -	-	B, -	D, 0	

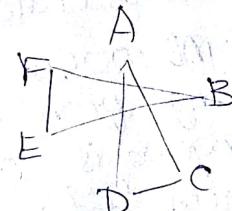
A	NS			
	00	01	11	10
A, 0	A, 0	B, -	A, 0	
B, 1	A, -	B, 1	B, 1	

y	NS			
	00	01	11	10
0	0 0 1 0			
1	0 0 0 1			

$$Y = DG + \bar{D}G$$

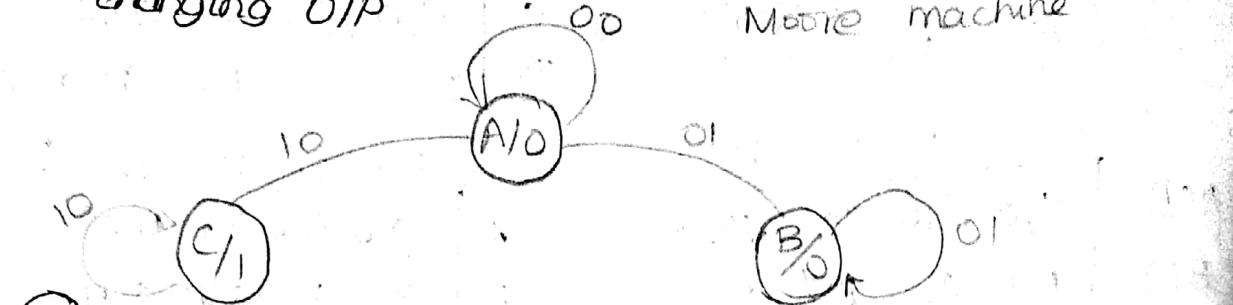
Excitation of SR

On	Q _{n+1}	S	R
0	0	0	X
0	1	1	0
0	1	0	1
1	1	X	X



- 3) Develop the state diag & primitive flow table for logic sim that has 2 I/P $x \& y$ & 1 O/P z , behaves in following manner. Initially both I/P & O/P are 0 whenever $x=1$ & $y=0$, z becomes 1 & whenever $x=0$ & $y=1$, z becomes 0. When $x \& y$ are 0/1, there's no change in $y=1$, $z=0$. When $x \& y$ are 0/1, there's no change in $y=1$, $z=0$.

O/P, O/I/O remains in prev. state. Logic S/no has edge triggered I/O without clk the logic sim changes state on the rising edge of 2 states. Static I/O values are not having any effect in changing O/I/O.



(4) Transition table \Rightarrow states in binary.

xy	AB	00	01	11	10
00	00	00	11	00	01
01	00	00	01	00	00
11	11	11	11	00	11
10	10	11	11	00	10

$\circlearrowleft \rightarrow$ stable state

$xyAB = 0000$ } oscillation
 $AB \rightarrow 10$ }

$AB \rightarrow$ input variable

$xy \rightarrow$ I/O state $xy = XY$

↑
stable

- * I/O state $\&$ equal to excitation variable \Rightarrow state
- * Change in I/O \Rightarrow change in excitation variable
- * Only one variable change in I/O \Rightarrow fundamental mode of operation

(1) $AB \rightarrow 10$ Excitation variable = 01.

Forces to make I/O variable to 01 (i.e) 00
 \therefore It oscillates b/w 01 & 00 (ZSR).

(2) ~~X~~ $xyAB : 0000$, $AB : 01$

Change from 00 to 11 is not instantaneous
The change is 00 to 01 to 11
(critical state)

When you have a change in AB to 11

$xyAB : 1110$. Able to go to stable state irrespective of path chosen. The delay involved is not giving any problem. This is non-critical race

Non-critical \Rightarrow All variables are same

~~X~~ Critical \Rightarrow All excitation variables are not same

	AB	00	01	11	10
xy	00	11	00	11	00
	01	01	11	11	01
	11	10	11	11	10
	10	10	10	11	11

An asynch sequential ckt the state variable x and y with primary I/O A and B & its own feedback x & y as shown in the K-map. Can this ckt face any problem in its operation.

In $CC = \text{Stable i/p} \Rightarrow \text{no feedback} \Rightarrow \text{No Hazard}$

Delay in path \Rightarrow critical \Rightarrow Asynchronous seq. ckt.

2 TYPES of Hazards \leftarrow static 1 \Rightarrow expectation is O/P

Expectation is O/P \leftarrow Static 0

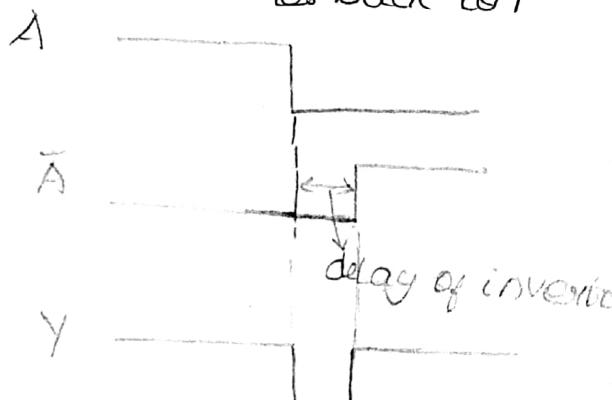
to be 0. But it momentarily it goes to 1 & comes back to 0

should be 1. But momentarily it changes 0 and then comes back to 1



	$\bar{A}\bar{B}$	\bar{C}	C
	0 0	0	0
	1 0	0	1
	1 1	1	1
	0 1	1	0

$$\Rightarrow \bar{B}\bar{C} + AC$$

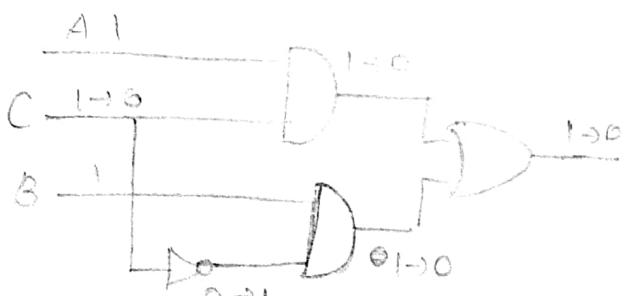


Static 1 hazard.

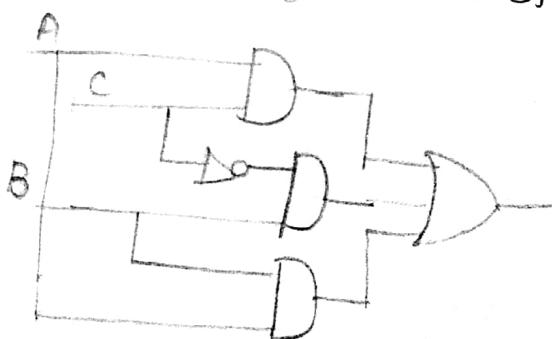
	\bar{C}	C
	0 0	0
	1 0	1
	1 1	1
	0 1	0

$$Y = \bar{B}\bar{C} + ACT + AB$$

SOP.



	$\bar{A}\bar{B}$	\bar{C}	C
	0 0	0	0
	1 0	0	1
	1 1	1	1
	0 1	1	0



$$\text{Expression in POS: } (A + \bar{C})(B + C)$$

Dynamic & Essential Hazard -

↓ Different path delay in feedback
ckt makes multiple transitions
ckt with f even when change in I/P
does not reach one point of ckt