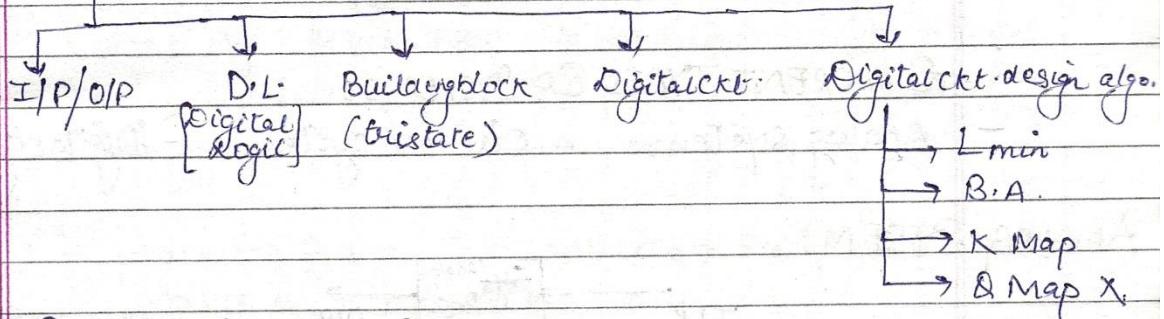


TOPICS TO BE COVERED

(1) Intro



(2) Combinational circuit

(3) Sequential circuit

(4) State Machine

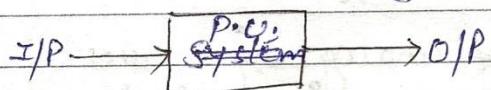
(5) MSI / LSI

(6) logic family

(7) ADC / DAC

SYSTEM & CIRCUIT:

SYSTEM: Every system has strictly 3 units:
I/P, processing unit, O/P



I/P may be energy, material or information &
O/P also may be energy, material or information

SIGNAL: information carrier

e.g. population of a city

$$f(t) = \sin t$$

$$v(t) = \sin t v$$

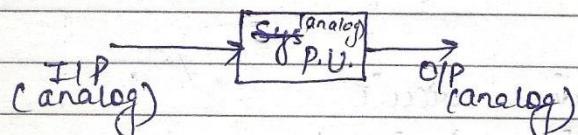
$$f(t) = \sin t \cdot v$$

$$\int_{v_0}^v$$

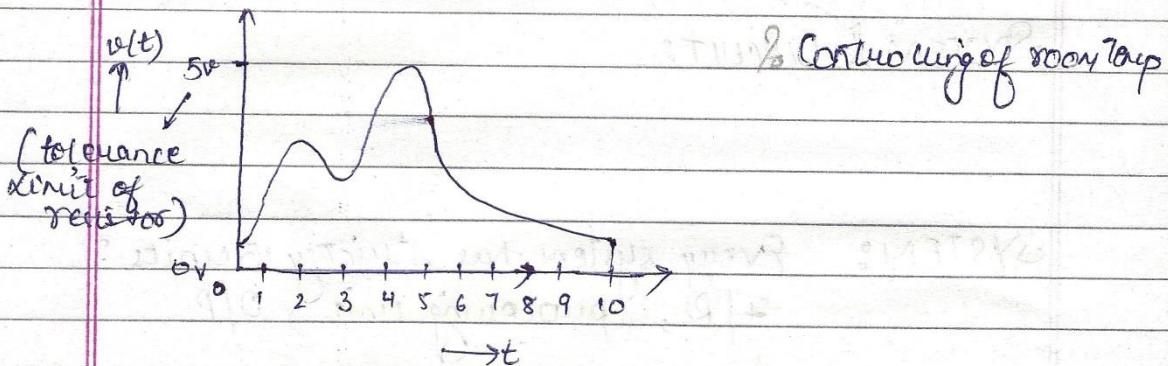
DIFFERENT TYPES OF SYSTEMS:

- Analog System
- Discrete System
- Digital System

ANALOG SYSTEM:



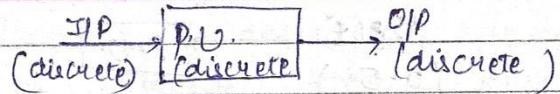
- All natural signals are basically analog signals
- Analog signal can take any value at any instant of time (independent variable) within the range specified.



drawbacks of analog signal are:-

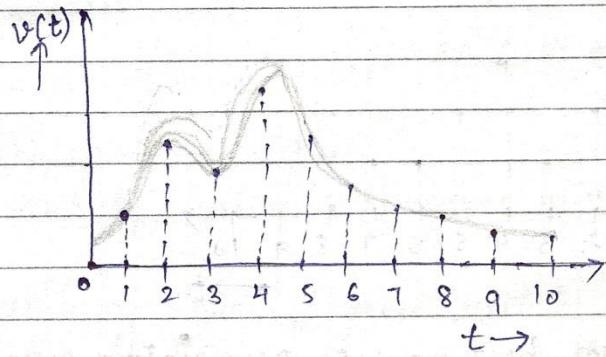
- (i) storage of analog signals is impossible (\because of infinite no. of points within two points)
- (ii) unreliable system
- (iii) corrupted data / noise effect is more and exists till end.

DISCRETE SYSTEMS:



In this type of system, signal exists at particular instant of time not in between. i.e. there exists discretization w.r.t. independent variable.
This process is known as sampling process.

Sampling process - process of collecting samples at discrete value of independent variable.



- No. of data has been minimised ^{than} that of analog.
- storage is easy than that of analog system.
- effect of noise exists but at particular instant.

drawbacks of discrete systems are:-

5 V - 200 V

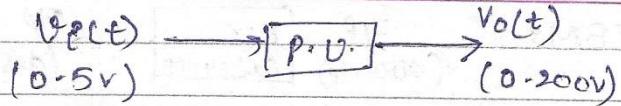
4.9 V - 199 V

4.8 V - 198 V

Hardly any difference

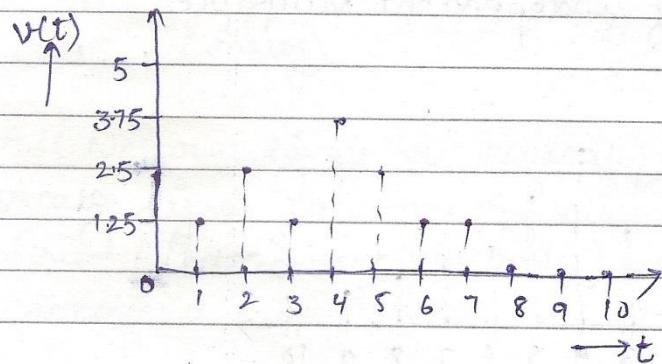
DIGITAL SYSTEMS:- Process of discretisation w.r.t. dependent variable.

Quantisation - process of discretisation w.r.t. dependent variable.



$V_i(t)$ regd.	$V_i(t)$	$V_o(t)$	$V_o(t)$ regd.
0 v	0-1.25	0-50v	0 v
1.25v	1.25-2.5	50-100v	50 v
2.5 v	2.5-3.75	100-150v	100 v
3.75 v	3.75-5	150-200v	150 v

equally
9

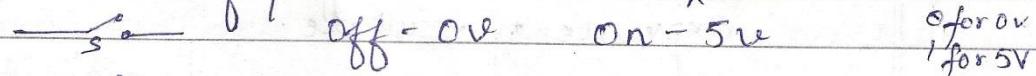


it is digital but values associated are analog
Processing cannot be done

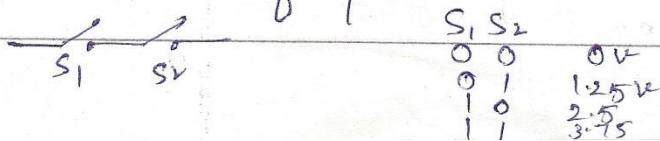
ENCODING PROCESS: This process is used to convert analog to digital code.

To encode 2^n levels of quantisation we have to assume n switches and that will result in n bit digital data.

e.g. for 2 levels of quantisation switches are 1



e.g. for $4^{(2^2)}$ levels of quantisation switches requires



e.g. To generate 5 levels of quantization we will use 3 switches but we will take 3 levels/stages as don't care.

Advantages of digital system
easy storage, less effect of noise.

- Accuracy & performance can be improved in both the case analog & digital by designing more efficient hardware. But it is easier to design more efficient H/W in case of digital systems.
- Analog → digital
cost of system increases but performance becomes better.
- Increasing levels of quantization improves performance
e.g. In 8 levels till 175V O/P can be obtained
- All digital signals are analog.
- Cost is important or under consideration upto sequential ckt's.

Analog I/P

P.U. (analog)

Analog O/P

ADC

Digital I/P

P.U. (digital)

Digital O/P

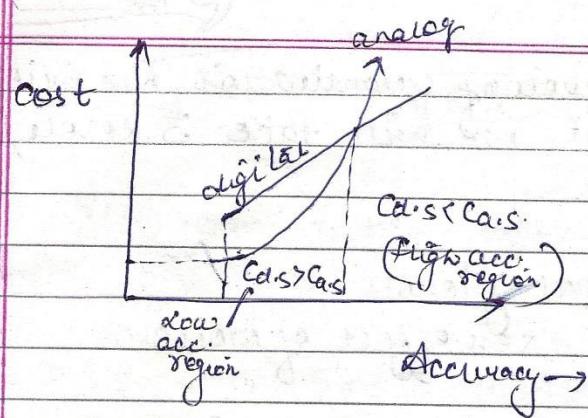
DAC

Analog O/P

10. In digital system

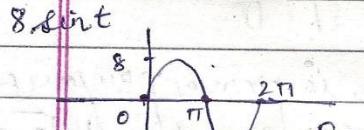
two extra units are required.

Mega



e.g. Voltage signal $v(t) = 8 \sin(\frac{2\pi}{8}t)$ is digitised with sampling interval of 1 sec and 3 bit quantiser. Then the digital data associated with digital signal at $t=1$ sec, 2 sec, 3 sec, 4 sec.

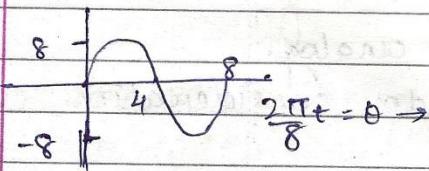
$$v(t) = 8 \sin\left(\frac{2\pi}{8}t\right) \quad 0 < t \leq 4 \text{ sec.}$$



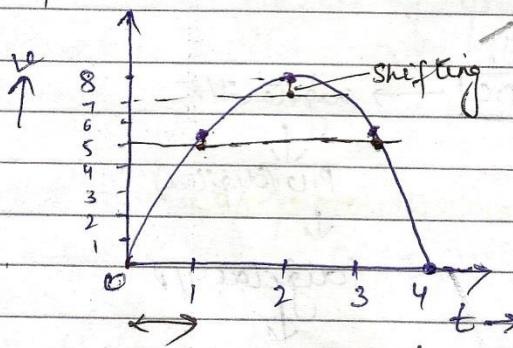
$$\text{In Ques. } \theta = \frac{2\pi t}{8}$$

$$\text{If } \theta = \pi \text{ then } \theta' = 4$$

$$\theta = 2\pi \text{ then } \theta' = 8$$



Therefore $0 < t \leq 4 \text{ sec.}$



at $t = 1$ sec.

$$v(1) = 8 \times \frac{1}{\sqrt{2}} = 5.6 \text{ v}$$

$$v(2) = 8(1) = 8 \text{ v}$$

$$v(3) = 8 \times \frac{1}{\sqrt{2}} = 5.6 \text{ v}$$

$$v(4) = 8 \times 0 = 0 \text{ v}$$

$$f_s = 1/T_s$$

(13)
14

Quantisation interval

$$q = \frac{\text{max. value}}{\text{level of quantisation}} \quad \text{or} \quad q = \frac{\text{Full scale value}}{2^n}$$

- In numerical Pb.

$$q = \frac{8}{8} = 1$$

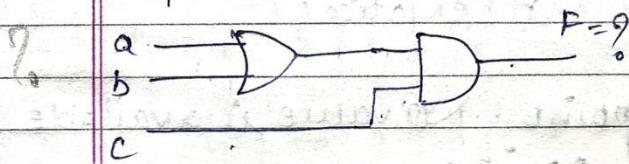
Ans at $t=1$ sec digital data associated = 5 = 0101

at $t=2$ sec " " " = 7 = 10000111

at $t=3$ sec " " " = 5 = 0101

at $t=4$ sec " " " = 0 = 0000

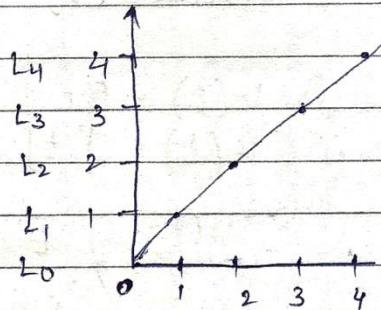
Application



QUANTISATION PROCESS

$$V(t) = t \quad 0 < t \leq 4 \text{ V}$$

$$q = \frac{4}{4} = 1$$



Two types of quantisation process:

- a) truncated quantisation
- b) rounding off quantisation

truncation - remove decimal fig

e.g. truncation upto 1 decimal place
of 5.47 is 5.4

L_i < a ≤ L_{i+1}

$$d_q = L_i$$

lower level truncated quantization

$$d_q = L_{i+1}$$

higher level truncated quantization

rounding off quantization

$$L_i < a \leq L_{i+1/2}$$

$$d_q = L_i$$

$$L_{i+1/2} < a < L_{i+1}$$

$$d_q = L_{i+1}$$

- in case of round off quantization, quantization error $q_e \rightarrow 0 \leq q_e \leq 0.5$

in case of truncated quantization $0 \leq q_e \leq 1$

but still we prefer truncated quantization
as error is mathematical

Offset - terminal point. No value is available at that point.

Quantization is with lower level or zero offset

C.Q.8

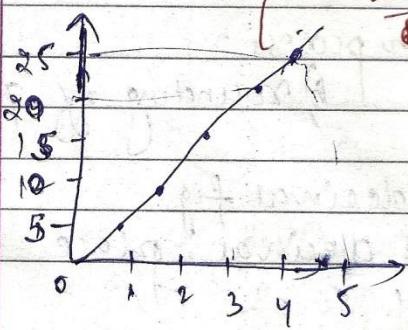
$$\text{i)} v(t) = 5t \quad 0 < t < 5 \text{ sec.}$$

$$\text{ii)} v(t) = [t] \quad 0 < t \leq 5 \text{ sec.}$$

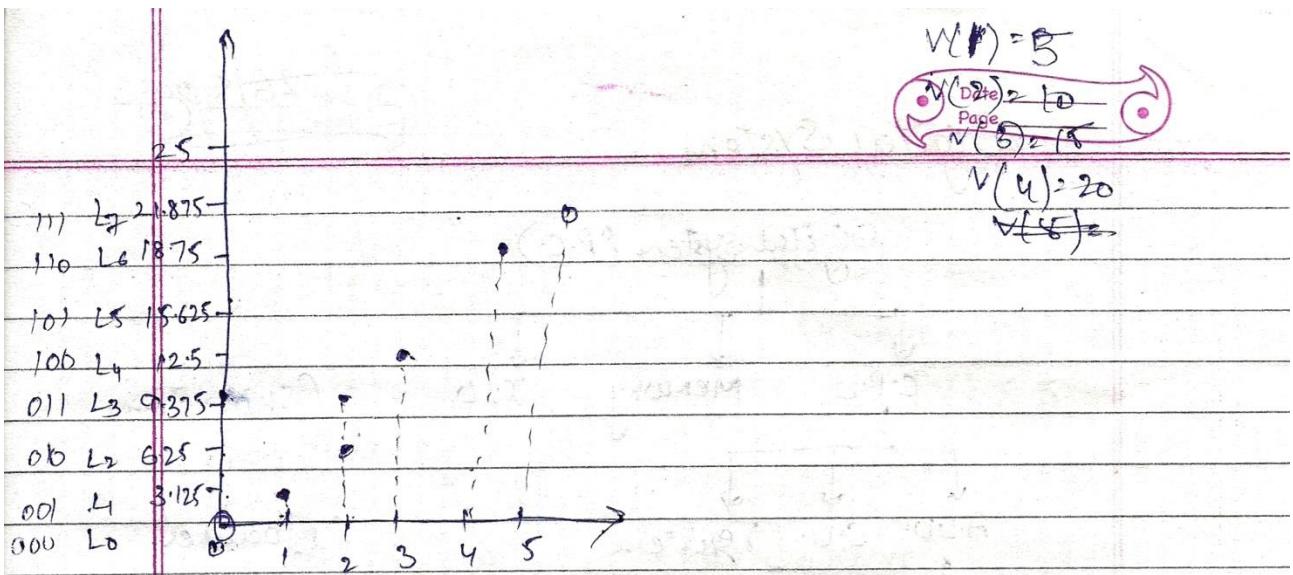
$$\text{iii)} v(t) = \int t \quad 0 \leq t \leq 2 \text{ sec.}$$

$$\frac{8 \sin 2\pi t}{8} \quad 2 < t \leq 4 \text{ sec.}$$

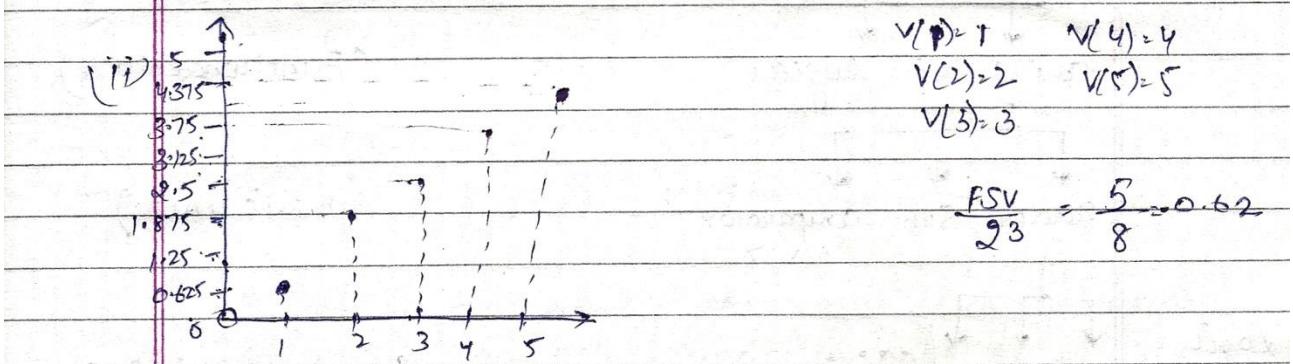
(D)



$$\begin{aligned} \text{ESV} &= \frac{25}{2^8} \\ &= \frac{25}{256} \\ &\approx 0.097 \end{aligned}$$



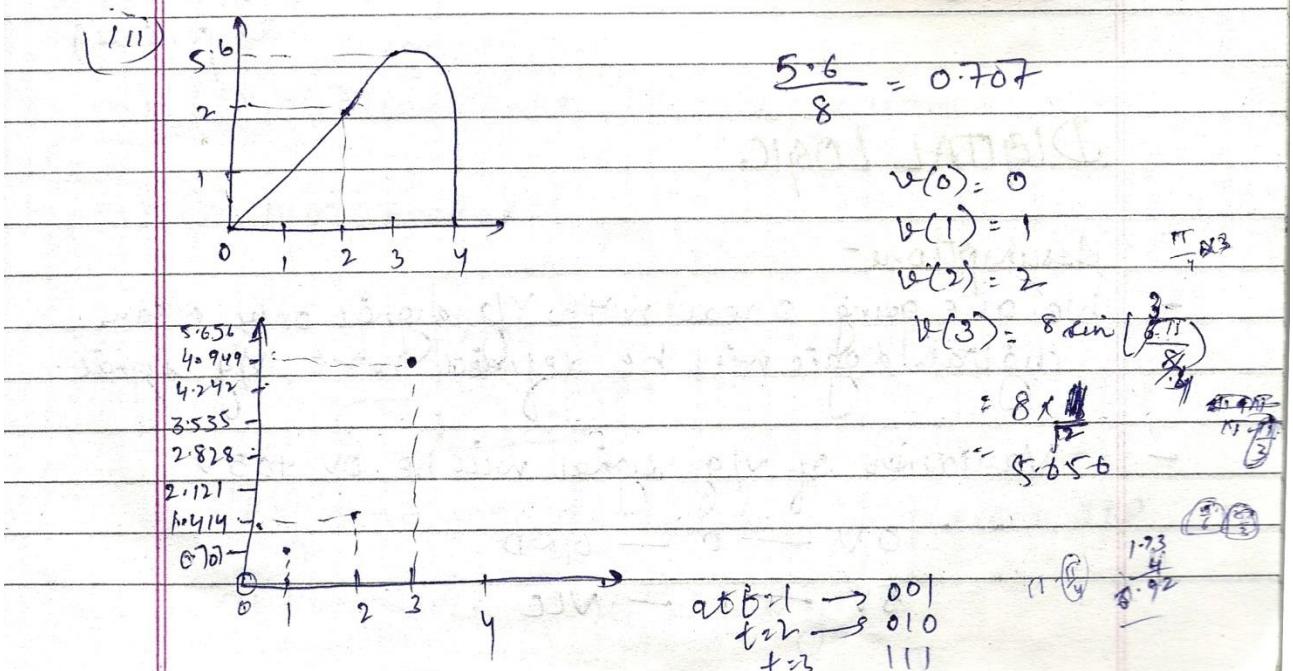
$$\begin{aligned}
 V(1) &= 5 \\
 V(2) &= 10 \\
 V(3) &= 15 \\
 V(4) &= 20 \\
 V(5) &= 25
 \end{aligned}$$



$$\begin{aligned}
 V(1) &= 1 & N(4) &= 4 \\
 V(2) &= 2 & V(5) &= 5 \\
 V(3) &= 3
 \end{aligned}$$

$$FSV = \frac{5}{23} = \frac{5}{8} = 0.62$$

$t=1 \rightarrow 001, t=2 \rightarrow 011, t=3 \rightarrow 100$
 $t=4 \rightarrow 110, t=5 \rightarrow 111$



$$\frac{5.6}{8} = 0.707$$

$$v(0) = 0$$

$$v(1) = 1$$

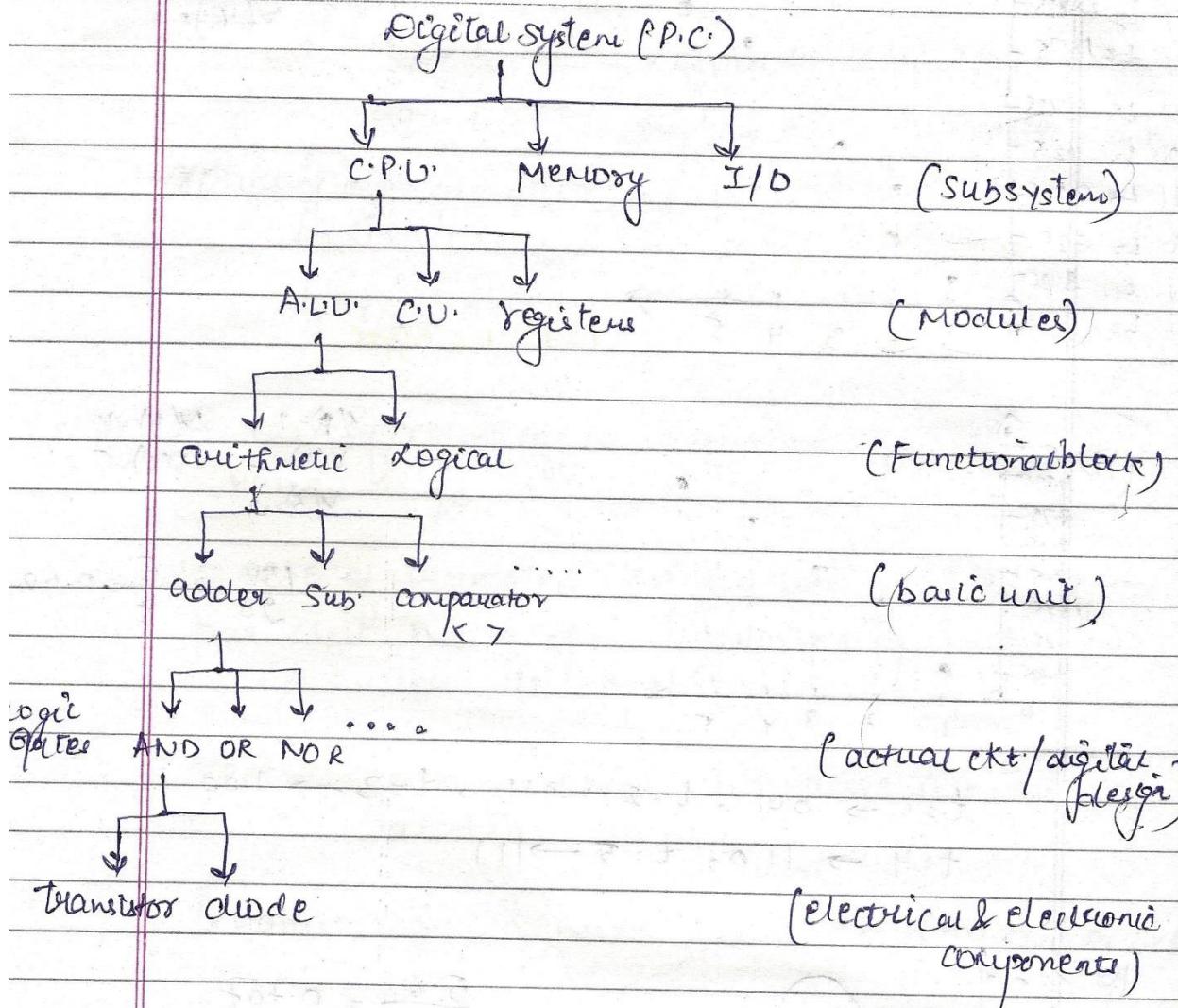
$$v(2) = 2$$

$$v(3) = 8 \sin\left(\frac{\pi}{8}\right)$$

$$= 8 \times \frac{\sqrt{2}}{2} = 5.656$$

at $t=1 \rightarrow 001$
 $t=2 \rightarrow 010$
 $t=3 \rightarrow 111$

Digital System



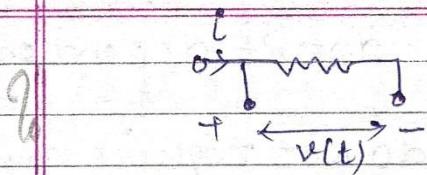
DIGITAL LOGIC

Assumptions -

- We are going to deal with V/g signals only means digital logic will be defined w.r.t V/g signals.
- Variation of V/g signal will be 0V to 5V

0V — 0 — GND

5V — 1 — Vcc



$v(t)$

$0 - 5V$

$0V - 5V = 0$

$$0 \leq |v| \leq 5$$

Two Types of logic:

Positive logic

$0 - 5V$

$0V - 0$

$5V - 1$

Negative logic

$0V - 1$

$5V - 0$

$-5V - 0V$

$-5V - 0$

$0V - 1$

$-5V - 1$

$0V - 0$

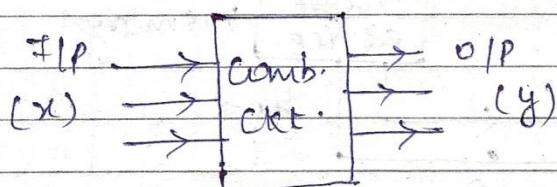
In case of +ve logic higher value of input variable represents logic 1 whereas in case of -ve logic higher value of I/P variable represents logic 0.

DIGITAL CKT:

a) Combinational ckt.

b) Sequential ckt.

a) Combinational ckt:



O/P of comb. ckt. depends only on present I/P

at $t = n\tau$ sec.

$$x_n \rightarrow y_n$$

O/P of combinational ckt. doesn't depend on history of system.

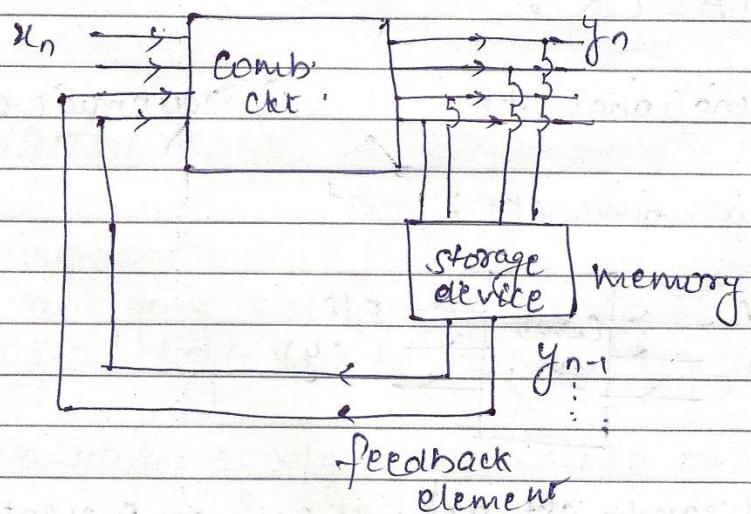
i.e. combinational ckt. do not require any storage element.

b) Sequential Circuit:

- O/P of sequential ckt depend on present I/P as well as past O/P
- Sequential ckt. depend on history. requires memory



$$\begin{matrix} x_n \rightarrow y_n \\ y_{n-1} \\ y_{n-2} \end{matrix}$$



Hence sequential ckt. is basically a comb. ckt. with storage device.

BUILDING BLOCKS

Basic blocks which construct digital ckt.

7 gates:

- | | | |
|--------------------|----------------------|---|
| Basic gates | - AND | } Fundamental gates

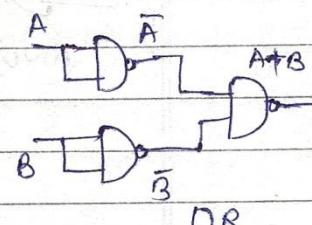
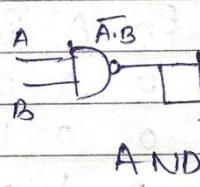
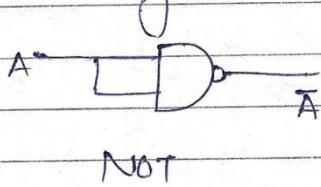
} Universal gate |
| | - OR / inclusive OR | |
| | - NOT / inverter | |
| | - NOR | |
| | - NAND | |
| | - XOR / exclusive OR | |
| | - XNOR / EX-NOR | |

Any digital logic can be implemented by using proper combination of fundamental gates.

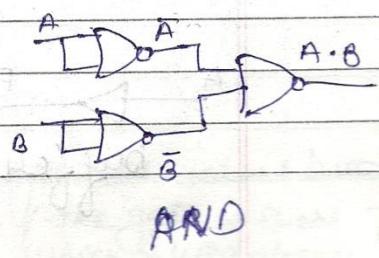
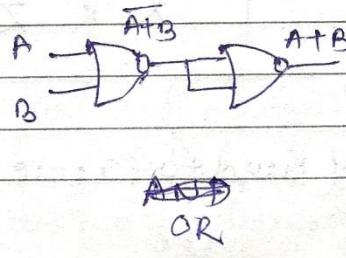
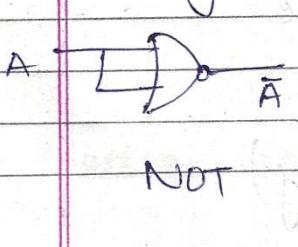
Any digital logic can be implemented by using NAND gate alone or NOR gate alone.

Pb Implement AND, OR, inverter gate by using
a) NAND gate alone b) NOR gate alone.

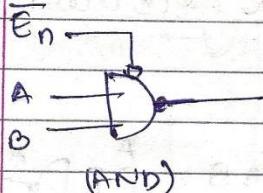
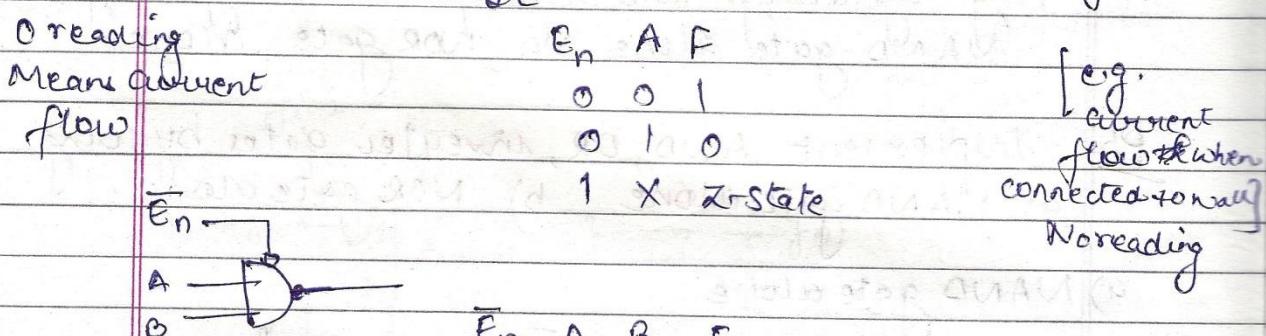
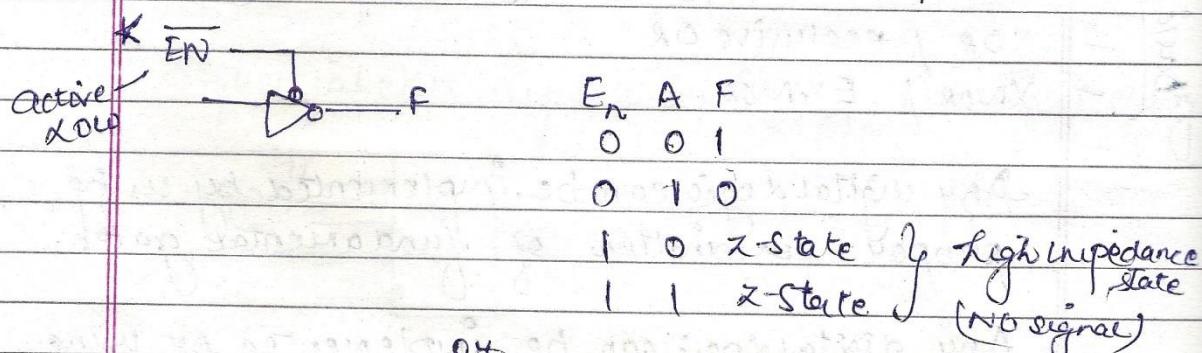
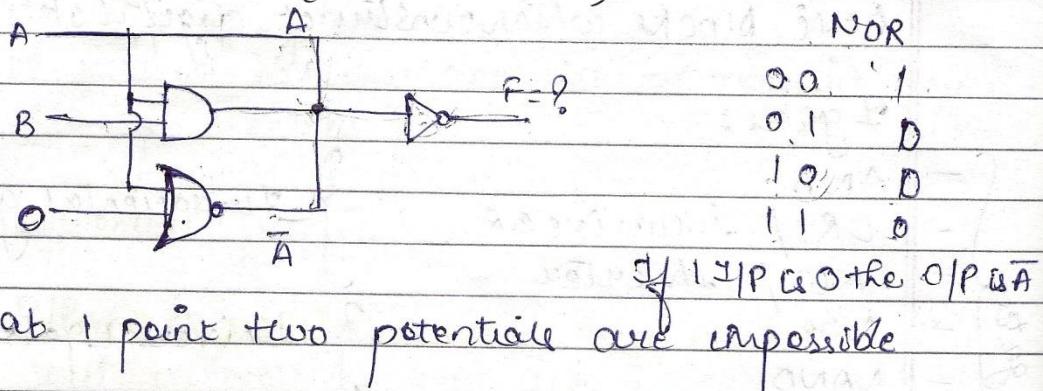
a) NAND gate alone



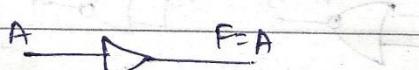
b) NOR gate alone



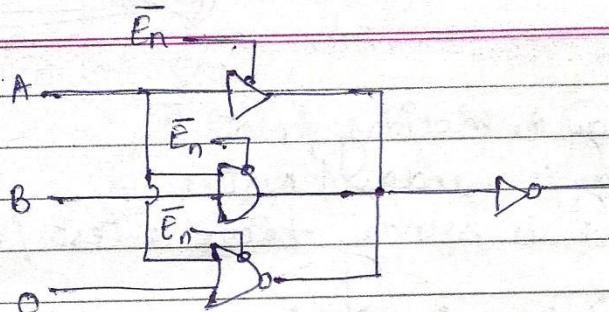
TRISTATE GATE (3 STATE GATE):



EN	A	B	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	X	X	Z-state

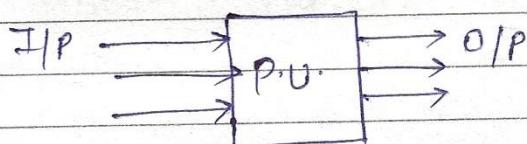


buffer (used to cause delay)



t	1	2	3
E _{n1}	0	1	1
E _{n2}	1	0	1
E _{n3}	1	1	0

COMBINATIONAL CIRCUIT DESIGN ALGO.



If I/P & P.U. are known then it is known as Analysis

If I/P & O/P are known then it is known as Design

PROBLEM STATEMENT

I/P O/P relation known

↓
logical expression

[I/P, O/P binary + relation]

↓
Truth Table

[I/P, O/P relation in table]

Canonical SOP

↓
Boolean Expression

[Mathematical representation of I/P]

Canonical POS

↓
minimization process

↓
Minimum logic

K-M.
B.A.

↓
Optimal soln.

↓
Implementation of optimal soln.

In most of cases minimum logic is optimal soln. but
Not always [minimum no. of gates reqd. / the gate is such which is used already]

Pb

e Constraints

A \rightarrow Today is Holiday / Not

B \rightarrow Today is Sunday / not

C \rightarrow If it is Sunday there is test / not

F \rightarrow If it is holiday or if there is Sunday and
there is no test then the day is free
from for watching movie

$$F = A + B \cdot \bar{C}$$

Min. Term	A	B	C	F	
M ₀ , $\bar{A}\bar{B}\bar{C}$	0	0	0	0	= 1
M ₁ , $\bar{A}\bar{B}C$	0	0	1	0	$0 \cdot 0 \cdot 1 = 0$
M ₂ , $\bar{A}B\bar{C}$	0	1	0	1	$0 \cdot 1 \cdot 0 = 0$
M ₃ , $\bar{A}BC$	0	1	1	0	$0 \cdot 1 \cdot 1 = 0$
M ₄ , $A\bar{B}\bar{C}$	1	0	0	1	
M ₅ , $A\bar{B}C$	1	0	1	1	
M ₆ , $AB\bar{C}$	1	1	0	1	
M ₇ , ABC	1	1	1	1	

Each min. term represents that state uniquely

A, B, C - I/P variable

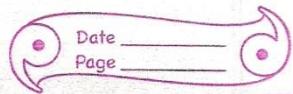
F - O/P variable

A, B, C, $\bar{A}\bar{B}\bar{C}$ - literals

If there are n inputs then there exists 2ⁿ literals

Each state is represented by mathematical term -
Min. term

(direct representation of TT)



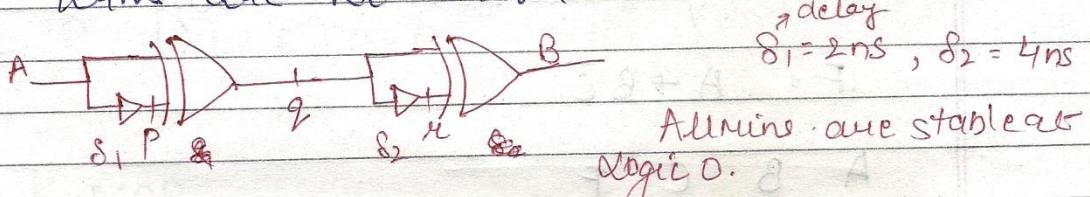
Canonical SOP =

$$\begin{aligned}
 & \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC \\
 & = m_2 + m_4 + m_5 + m_6 + m_7 \\
 & = \sum m_2, m_4, m_5, m_6, m_7 \\
 & = \sum m(2, 4, 5, 6, 7)
 \end{aligned}$$

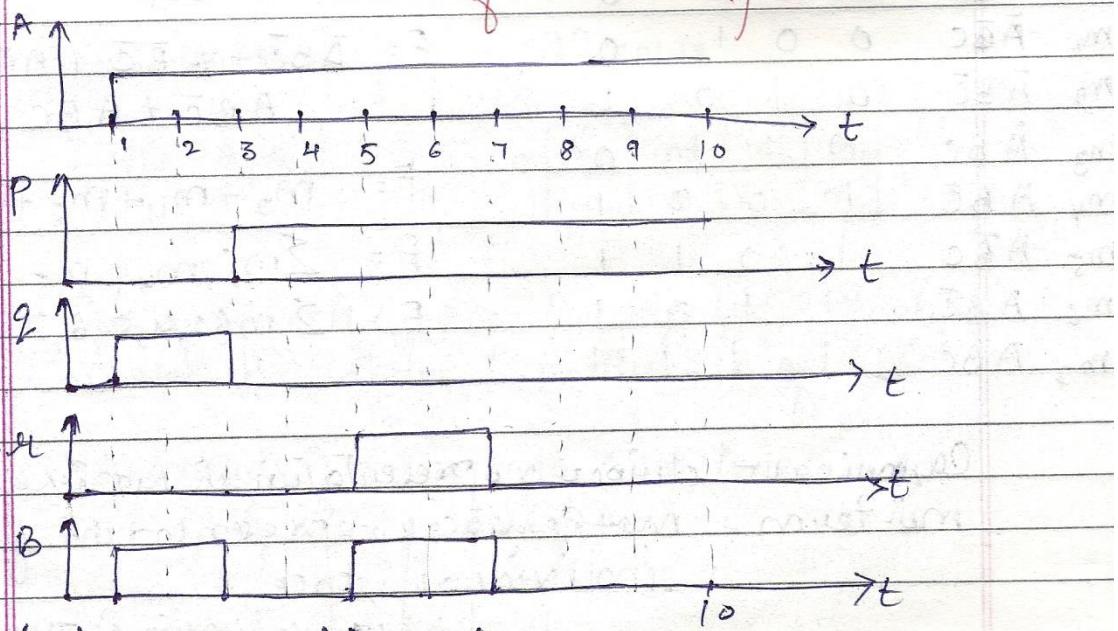
There exists difference between product term & min-term.
Min term represents uniquely single state of system / I/P variable.
and min term is associated with at least 1 literal of each variable.

All min terms are product terms but all product terms are not min terms.

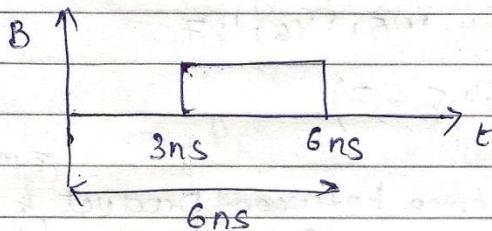
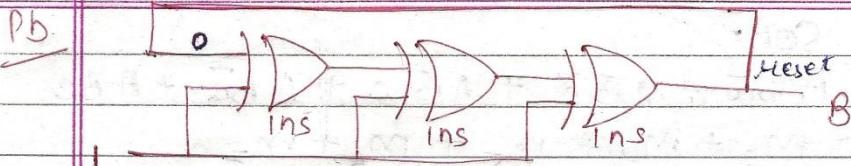
2/6/13
Pb



Find no of transitions from 0 to 1 ones at B.



4 transitions fill 0 to 1 ns



Pb We need a S/W through which we can predict how many MBA aspirants are going to take education loan

A — candidate is qualified in CAT / not

B — " " " GMAT / Not

C — financial aid availability

$$F = A + B\bar{C}$$

$$\begin{array}{cccc} A & B & C & F \end{array}$$

m_0	$\bar{A}\bar{B}\bar{C}$	0	0	0	0	Canonical SOP
m_1	$\bar{A}\bar{B}C$	0	0	1	0	$F = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}C$
m_2	$\bar{A}B\bar{C}$	0	1	0	1	$A\bar{B}\bar{C} + ABC$
m_3	$\bar{A}BC$	0	1	1	0	
m_4	$A\bar{B}\bar{C}$	1	0	0	1	$F = m_2 + m_4 + m_5 + m_6 + m_7$
m_5	$A\bar{B}C$	1	0	1	1	$F = \sum m_2, m_4, m_5, m_6, m_7$
m_6	$AB\bar{C}$	1	1	0	1	$F = \sum m(2, 4, 5, 6, 7)$
m_7	ABC	1	1	1	1	

Canonical — direct representation in mathematical form
 minterm — mathematical representation of individual states
 (combination of IP variables)

Identities

$$A + \bar{A}B = A + B$$

$$\bar{A} + AB = \bar{A} + B$$

De Morgan

$$\bar{A} \cdot \bar{B} = A + \bar{B}$$

$$\bar{A} + \bar{B} = A \cdot \bar{B}$$

$$\begin{aligned}
 F &= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}C \\
 &= \bar{B}\bar{C}(\bar{A} + A) + A\bar{B}(\bar{C} + C) + ABC \\
 &= \bar{B}\bar{C} + A\bar{B} + ABC = B(\bar{C} + AC) + A\bar{B} \\
 &\cancel{=} \bar{B}\bar{C} + A(B + BC) = B(\bar{C} + A) + A\bar{B} \\
 &\cancel{=} \bar{B}\bar{C} + A(\bar{B} + C) = AB + B\bar{C} + A\bar{B} \\
 &\qquad\qquad\qquad A(B + \bar{B}) + B\bar{C} \\
 &\qquad\qquad\qquad A + B\bar{C}
 \end{aligned}$$

% $\bar{A}\bar{B}\bar{C} + A + \textcircled{a}$ \bar{ABC}
Dominance

$$P = Q + R$$

$$P + Q = P$$

$$P + R = P$$

$$P + Q + R = P$$

$$P + Q + Q + R + R = P$$

P	Q	R	$P + Q$	$P + R$	$P + Q + R$	$P + Q + R$
0	0	0	0	0	0	0
0	1	0	1	0	1	1
1	0	0	1	1	1	1
1	1	0	1	1	1	1

K-MAP

If n IIP then K-map is of 2^n cells

3IIPs A, B, C

	BC	00	01	11	10
A		m_0	m_1	m_3	m_2
	1	m_4	m_5	m_2	m_6

- Coding of cell is acc. to gray code. In b/w consecutive states transition is of 1 bit

- Numbering of cell acc. to min. term

	BC	00	01	11	10
		0	1	3	2
	1	4	5	7	6

- binary method is not used in cell coding
" " problem in pairing (Logical + geometrically consecutive)
- implicants (in k-map) & minterms (in TT)
are synonymous
i.e. implicants are of individual cell &
min terms " " state

→ Pairing is done acc. to 2^n

$$\begin{array}{ll} 2^0 - 1 & 2^2 - 4 \\ 2^1 - 2 & 2^3 - 8 \end{array}$$

Pairing must be
of w consecutive cells

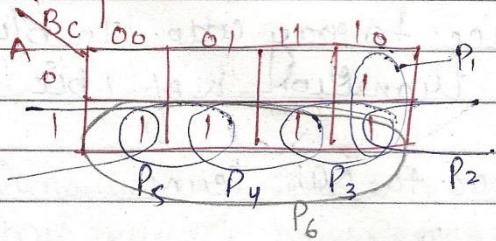
- Try max. No. of cells pair
- Overlapping is allowed

→ Implication of More than 1 cell are known as
priniciple implicants

- Unchanged parameter + its value is used for representation

To - designer dependency exists

P5
— Find No. of essential & non-essential prime
implicants



$$\begin{aligned} F &= P_1 + P_6 \\ F_1 &= P_1 + P_4 + P_5 \\ F_2 &= P_1 + P_2 + P_4 \\ F_3 &= P_1 + P_3 + P_5 \\ F_4 &= P_1 + P_6 \end{aligned}$$

Essential P.I. - P_1 , $\ell = 1$

Non essential P.I. - P_2, P_3, P_4, P_5, P_6 = 5

→ Minimum logic is that one from all possible solutions in which Min. No. of ~~per~~ non-essential prime implicants are included.

Hence minimum logic $F = P_1 + P_6$

→ If there is overlapping in cell then it is Non essential P.I. Otherwise it is E.P.I.

POS form

$$\overline{F} = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{ABC}$$

$$\overline{\overline{F}} = F$$

$$\begin{aligned} \therefore F &= \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{ABC} \\ &= \overline{\overline{ABC}} \cdot \overline{\overline{A}\overline{B}C} \cdot \overline{\overline{A}\overline{B}\overline{C}} \end{aligned}$$

Canonical POS form $F = (A+B+C)(A+B+\bar{C})(A+\bar{B}+\bar{C})$

Max-term M, another unique representation of individual terms

$$F = TIM(0,1,3)$$

$$\begin{aligned} &= (A \cdot A + AB + A\bar{C} + BA + B \cdot B + B\bar{C} + CA + CB + C \cdot \bar{C})(A + \bar{B} + \bar{C}) \\ &= (A + AB + A\bar{C} + B\bar{A} + B + B\bar{C} + CA + CB + 0)(A + \bar{B} + \bar{C}) \\ &= (A(1+B) + A\bar{C} + B(1+\bar{C}) + CA + CB)(A + \bar{B} + \bar{C}) \\ &= (A + A\bar{C} + B + CA + CB)(A + \bar{B} + \bar{C}) \\ &= A + A(C + \bar{C}) + B(1 + C)(A + \bar{B} + \bar{C}) \\ &= (A + B)(A + \bar{B} + \bar{C}) \end{aligned}$$

$$\begin{aligned}
 & A + AB + A\bar{C} + AB + 0 + B\bar{C} \\
 & A(\Rightarrow 1 + \bar{B}) + \bar{B}A\bar{C} + AB + B\bar{C} \\
 & A + A\bar{C} + AB + B\bar{C} \\
 & A + AB + B\bar{C} = A + B\bar{C}
 \end{aligned}$$

	00	01	11	10
0.	0	0	0	1
1	1	1	1	1

$$F = \bar{A}\bar{B} + \bar{A}C \Rightarrow F = (A+B)(A+\bar{C})$$

In each minterm, if there is 1 then sum of Q1/Q2/P1/P2
depends on 1. If there is 0 then it depends on 0

in Q1/P1 for AND
in Q2/P2 for OR

PB (i) $F(P,Q,R,S) = \sum m(0,2,3,7,11,13,14,15)$

(ii) $F(A,B,C,D) = \sum m(2,6,7,9,13,15)$

(iii) $F(W,X,Y,Z) = \sum m(0,2,3,5,7,8,10,11,14,15)$

	00	01	11	10
00	1	0	1	1
01	4	5	1	6
11	12	13	1	14
10	8	9	1	10

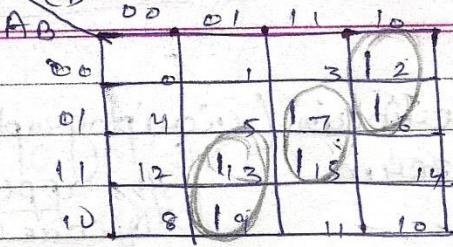
RS + $\bar{P}\bar{Q}\bar{S}$ + $PQST$
 PQR

Total prime implicants = 9

Essential = 3

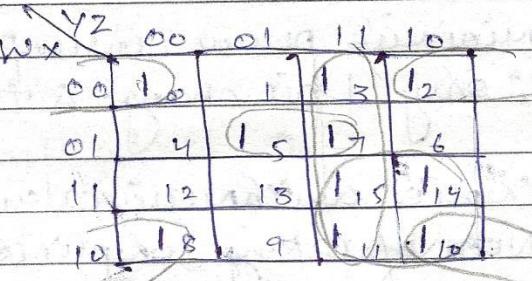
N.E.P.I. = $9 - 3 = 6$

Date _____
Page _____

(ii) 

$F = \bar{A}C\bar{D} + B\bar{C}D + A\bar{C}D$

Total P.I. = 5
Essential P.I. = 2
 $N.E.P.I. = 5 - 2 = 3$

(iii) 

$F = YZ + \bar{X}\bar{Z} + WY + \bar{W}XZ$

Total = 17 - 16
Essential = 1
 $N.E.P.I. = 17 - 1 = 16$

[More no. of pairs less is cost]

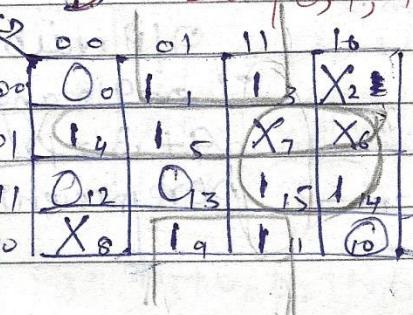
DON'T CARE STATEMENT [used for cost minimization]

Don't care - states combination of I/P variable for which don't worry about O/P

for SOP - assume these as 1 and for POS as 0

To include don't care is NOT MUST
representation for don't care - 2/X/d

Pb $D = \sum d(0, 4, 7)$ $D = \sum d(2, 6, 7, 8)$ $F = \sum m(1, 3, 4, 5, 9, 11, 14, 15)$



$F = \bar{A}D + \bar{A}B + \bar{B}C$

Plz ignore !

CODE CONVERTER

- Coding is done for inception (ryptography)
Password hidden form [S/w application of code conversion]
- H/w application
- gray code - to minimize power consumption
" only 1 bit change is there
- excess 3 code - simplifies arithmetic calculation.
(No need of taking complement)

Subtraction - a) borrow b) 2's complement

$$A - B = D = A + B\text{'s complement}$$

BCD code

No. of bits reqd. depend upon range

e.g.

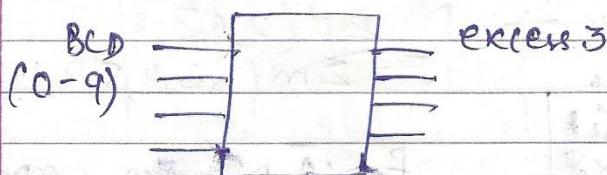
0-19

for 9 4 bits are reqd.

for 1 1 bit is reqd.

Total bits reqd. 5 bits

BCD to, excess 3



O/P will be of 4 bits
 $\because 0-9$ range
 $9+3=12$ can be represented in 4 bits

$$C_2 = \sum m(1, 2, 3, 4, 9)$$

$b_1 b_0$	00	01	11	10
$b_3 b_2$	*	*	*	*
00	*	1	1	1
01	1	*	5	7
11	X ₁₂	X ₁₃	X ₁₅	X ₁₄
10	8	19	X ₁	X ₁₆

$$e_2 = b_2 \bar{b}_1 \bar{b}_0 + b_0 \bar{b}_2 + \bar{b}_2 b_1$$

$$e_1 = \sum m(0, 3, 4, 7, 8)$$

$b_1 b_0$	00	01	11	10
$b_3 b_2$	*	*	*	*
00	1	1	1	1
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

$$e_1 = \bar{b}_1 \bar{b}_0 + b_1 b_0$$

$$e_0 = \sum m(0, 2, 4, 6, 8)$$

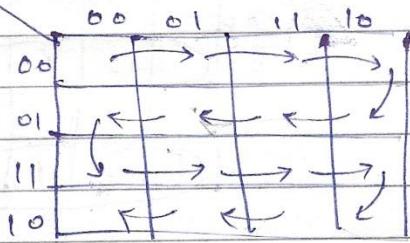
$b_1 b_0$	00	01	11	10
$b_3 b_2$	*	*	*	*
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

$$e_0 = \bar{b}_0$$

4 bit binary to gray code

Method to generate gray code

Move to consecutive cell in k-map



To generally
how we
used to do

$b_3 \ b_2 \ b_1 \ b_0$

$g_3 \ g_2 \ g_1 \ g_0$

	0 0 0 0	0 0 0 0	0 0 0 0
0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1	0 0 1 1
3	0 0 1 1	0 0 1 0	0 0 1 0
4	0 1 0 0	0 1 1 0	0 1 1 0
5	0 1 0 1	0 1 1 1	0 1 1 1
6	0 1 1 0	0 1 0 1	0 1 0 1
7	0 1 1 1	0 1 0 0	0 1 0 0
8	1 0 0 0	0 1 1 0	0 1 1 0
9	1 0 0 1	1 1 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1	1 1 1 1
11	1 0 1 1	1 1 1 0	1 1 1 0
12	1 1 0 0	1 0 1 0	1 0 1 0
13	1 1 0 1	1 0 1 1	1 0 1 1
14	1 1 1 0	1 0 0 1	1 0 0 1
15	1 1 1 1	1 0 0 0	1 0 0 0

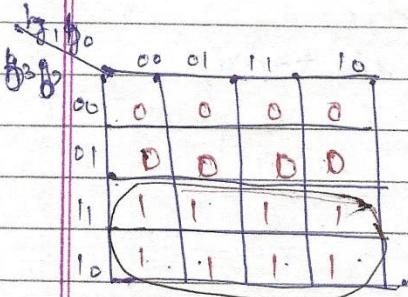
$$g_3 = \sum m(8, 9, 10, 11, 12, 13, 14, 15)$$

$$g_2 = \sum m(4, 5, 6, 7, 8, 9, 10, 11)$$

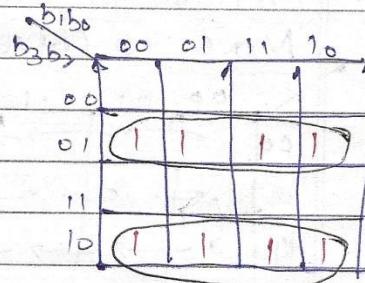
$$g_1 = \sum m(2, 3, 4, 5, 10, 11, 12, 13)$$

$$g_0 = \sum m(1, 2, 5, 6, 9, 10, 13, 14)$$

$$g_3 = \Sigma m(8, 9, 10, 11, 12, 13, 14, 15)$$



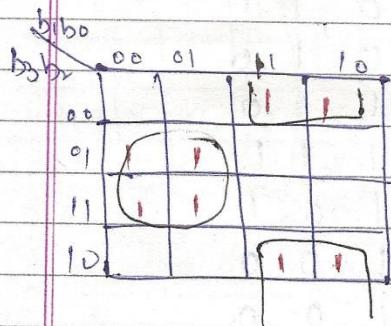
$$g_2 = \Sigma m(4, 5, 6, 7, 8, 9, 10, 11)$$



$$g_3 = b_3$$

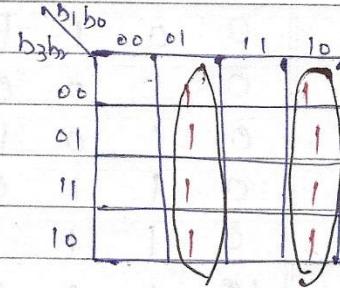
$$\begin{aligned} g_2 &= \bar{b}_3 b_2 + b_3 \bar{b}_2 \\ &= b_3 \oplus b_2 \end{aligned}$$

$$g_1 = \Sigma m(2, 3, 4, 5, 10, 11, 12, 13)$$



$$\begin{aligned} g_1 &= b_2 \bar{b}_1 + \bar{b}_2 b_1 \\ &= b_2 \oplus b_1 \end{aligned}$$

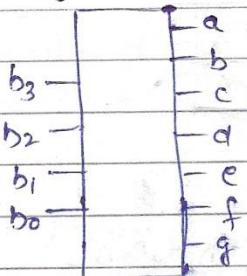
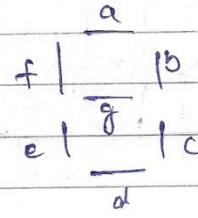
$$g_0 = \Sigma m(1, 2, 5, 6, 9, 10, 13, 14)$$



$$\begin{aligned} g_0 &= b_1 \bar{b}_0 + \bar{b}_1 b_0 \\ &= b_1 \oplus b_0 \end{aligned}$$

(0-9)

BCD to 7 Segment Display Decoder



	b ₃	b ₂	b ₁	b ₀	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	E
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	1	1	I

don't care

$$a_2 = \sum m (0, 2, 3, 5, 6, 7, 8, 9)$$

$$b_2 = \sum m (0, 1, 2, 3, 4, 7, 8, 9)$$

$$c_2 = \sum m (0, 1, 3, 4, 5, 6, 7, 8, 9)$$

$$d_2 = \sum m (0, 2, 3, 5, 6, 8, 9)$$

$$e_2 = \sum m (0, 2, 6, 8)$$

$$f_2 = \sum m (0, 4, 5, 6, 8, 9)$$

$$g_2 = \sum m (2, 3, 4, 5, 6, 8, 9)$$

$$D = \sum d (10, 11, 12, 13, 14, 15)$$

7-segment display output

$$A = \bar{b}_2 \bar{b}_0 + b_3 + b_1 + b_3 b_0$$

$$B = \bar{b}_2 + \bar{b}_1 \bar{b}_0 + b_1 b_0$$

$$C = \bar{b}_1 + b_0 + b_2$$

$$D = b_3 + \bar{b}_2 \bar{b}_0 + \bar{b}_1 b_0 b_2 + b_1 \bar{b}_2 + \bar{b}_3 \bar{b}_0$$

$$E = b_0 \bar{b}_2 + b_1 \bar{b}_0$$

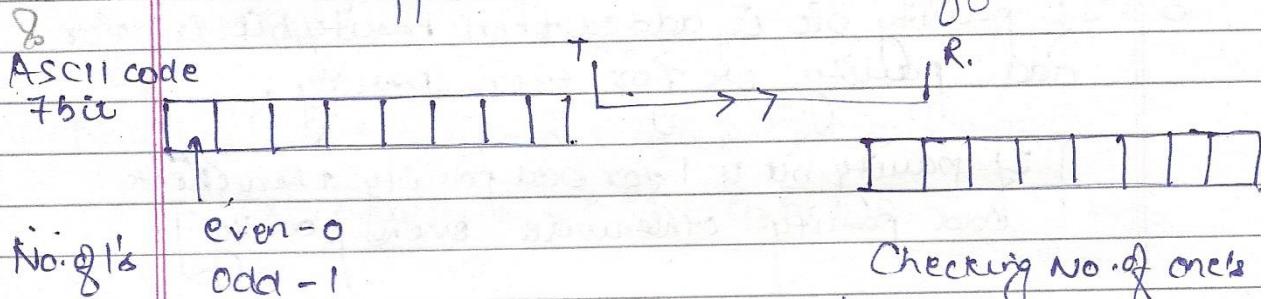
$$F = b_3 + b_2 \bar{b}_0 + b_1 \bar{b}_1 + \bar{b}_1 \bar{b}_0$$

$$G = b_3 + \bar{b}_1 b_2 + b_1 \bar{b}_0 + \bar{b}_2 b_2$$

PARITY CHECKER:

It is used to detect single bit error.

Parity checker & parity generator are same but application wise are different.



Checking No. of ones
If equal - No error
If not then error exists

→ Parity bit error & more than 1 bit error creates faulty situation

- If parity bit changes in O/P then it will give error but actually in data there is no error.
- If more than 1 bit error exists then no. of 1's will be as initial (odd/even) hence it won't give any error

→ Total no. of 1's in a bit state is known as parity bit

If No. of 1's is odd — odd parity
If No. of 1's is even — even parity

→ Well set of rules must be defined b/w transmitter & receiver end known as Protocol.

Hence protocol must be defined by

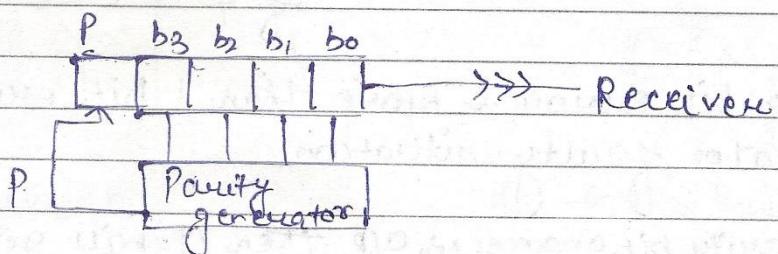
T & R ends :-

odd No. of 1's - 1/
Same in T & R

1. Total No. of bits going to receiver.
2. Whether parity bit is added or not
3. If parity bit is added then parity bit is 1 for odd parity or for even parity.

[If parity bit is 1 for odd parity then check odd parity otherwise even parity]

4 bit parity generator



	b_3	b_2	b_1	b_0	P	
0	0	0	0	0	0	odd parity=1, even p.=0
1	0	0	0	1	1	[odd parity generator]
2	0	0	1	0	1	
3	0	0	1	1	0	
4	0	1	0	0	1	$P = \sum m (1, 2, 4, 7, 8, 11, 13, 14)$
5	0	1	0	1	0	
6	0	1	1	0	0	
7	0	1	1	1	1	
8	1	0	0	0	1	
9	1	0	0	1	0	
10	1	0	1	0	0	
11	1	0	1	1	1	
12	1	1	0	0	0	
13	1	1	0	1	1	
14	1	1	1	0	0	
15	1	1	1	1	0	

$b_3 b_2$	00	01	11	10
00	0	1	3	12
01	14	5	7	6
11	12	13	15	14
10	18	9	11	10

B

$$F = \bar{b}_3 \bar{b}_2 \bar{b}_1 b_0 + \bar{b}_3 \bar{b}_2 b_1 \bar{b}_0 + \bar{b}_3 b_2 \bar{b}_1 \bar{b}_0 + \bar{b}_3 b_2 b_1 b_0 + \\ b_3 b_2 \bar{b}_1 b_0 + b_3 b_2 b_1 \bar{b}_0 + b_3 \bar{b}_2 \bar{b}_1 \bar{b}_0 + b_3 \bar{b}_2 b_1 b_0$$

$$= b_3 b_2 (\bar{b}_1 b_0 + b_1 \bar{b}_0) + \bar{b}_3 b_2 (\bar{b}_1 \bar{b}_0 + b_1 b_0) + b_3 b_2 (\bar{b}_1 b_0 + b_1 \bar{b}_0) + \\ + b_3 \bar{b}_2 (\bar{b}_1 \bar{b}_0 + b_1 b_0)$$

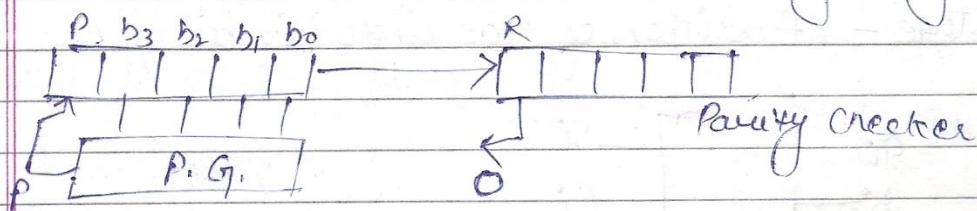
$$= \bar{b}_3 \bar{b}_2 (b_1 \oplus b_0) + \bar{b}_3 b_2 (\bar{b}_1 \oplus b_0) + b_3 b_2 (b_1 \oplus b_0) + \\ + b_3 \bar{b}_2 (\bar{b}_1 \oplus b_0)$$

$$= (b_1 \oplus b_0) [\bar{b}_3 \oplus b_2] + (\bar{b}_1 \oplus b_0) (b_3 \oplus b_2)$$

$$F = b_1 \oplus b_0 \oplus b_3 \oplus b_2$$

→ In receiver No. of 1's will be even when there is no noise. ∴ O/P will be generated i.e. NO ERROR

In case of parity checker 5 bits are considered including Parity bit



COMBINATIONAL CIRCUITS

Date 9/6/13
Page

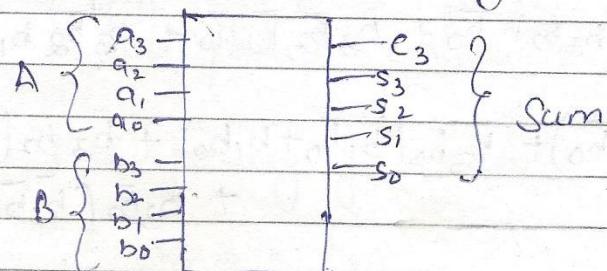
(ii) ARITHMETIC CIRCUITS

Nos. in
terms of
binary

$$A \rightarrow a_3 a_2 a_1 a_0$$

$$B \rightarrow b_3 b_2 b_1 b_0$$

Calculation of 4 bits may result in 5 bits



$$a_0 + b_0 \rightarrow c_0 s_0$$

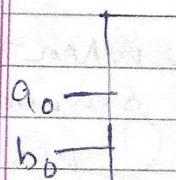
c_0 - carry bit
 s_0 - sum bit

$$a_1 + b_1 + c_0 \rightarrow c_1 s_1$$

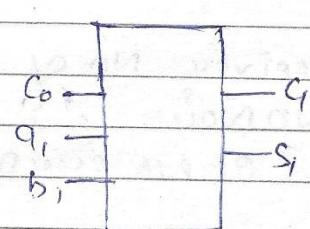
$$a_2 + b_2 + c_1 \rightarrow c_2 s_2$$

$$a_3 + b_3 + c_2 \rightarrow c_3 s_3$$

$$\begin{array}{r} c_2 \quad c_1 \quad c_0 \\ \overline{a_3 \quad a_2 \quad a_1 \quad a_0} \\ + b_3 \quad b_2 \quad b_1 \quad b_0 \\ \hline c_3 \quad s_3 \quad s_2 \quad s_1 \quad s_0 \end{array}$$

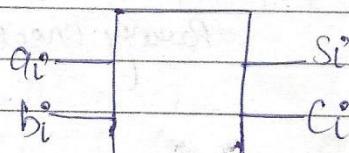


Ist stage



IIInd stage

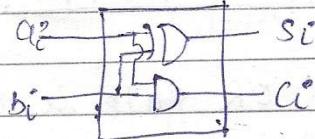
Ist Stage. - Addition of two single bit binary No.



$a_i \ b_i \ S_i \ C_i$

0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

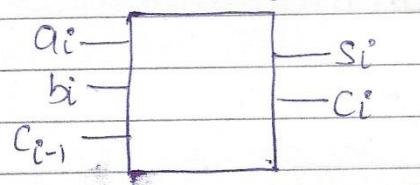
$S_i \rightarrow a_i \oplus b_i, C_i \rightarrow a_i \cdot b_i$



Half Adder

II Statement.

Addition of 3 single bit binary NO



LSB MSB

$a_i \ b_i \ c_{i-1} \ S_i \ C_i$

0	0	0	0	0
---	---	---	---	---

for S_i

1	0	0	1	0
---	---	---	---	---

2	0	1	0	1	0
---	---	---	---	---	---

3	0	1	1	0	1
---	---	---	---	---	---

4	0	1	0	1	0
---	---	---	---	---	---

5	1	0	1	0	1
---	---	---	---	---	---

6	1	1	0	0	1
---	---	---	---	---	---

7	1	1	1	1	1
---	---	---	---	---	---

$$S_i = a_i b_i c_{i-1} + \bar{a}_i \bar{b}_i \bar{c}_{i-1}$$

$$+ a_i \bar{b}_i \bar{c}_{i-1} + a_i b_i \bar{c}_{i-1}$$

$$C_i = \bar{a}_i b_i c_{i-1} +$$

$$a_i \bar{c}_{i-1} + a_i b_i$$

$$= C_{i-1}(a_i + b_i) + a_i b_i$$

↗ Minimum Logic

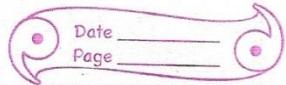
Now to check for
Optimal Soln. 7

$$= \bar{a}_i (\underline{b_i \oplus c_{i-1}}) + a_i (\bar{b}_i \oplus c_{i-1})$$

$$= \bar{a}_i P + a_i \bar{P}$$

$$= a_i \oplus P = a_i \oplus b_i \oplus c_{i-1}$$

Optimal soln - 9



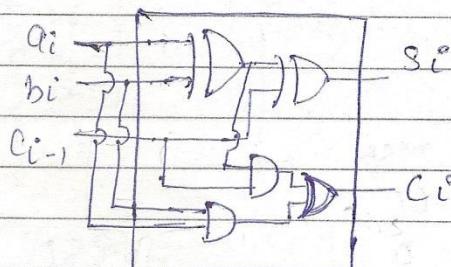
a_i	00	01	11	10
0	0	1	(13)	2
1	4	5	15	16

$$a_i b_i + a_i \bar{b}_i c_{i-1} + \bar{a}_i b_i c_{i-1}$$

$$a_i b_i + c_{i-1} (a_i \oplus b_i)$$

X-OR gate is heavy than OR but it is already used in s_i hence it is being implemented.

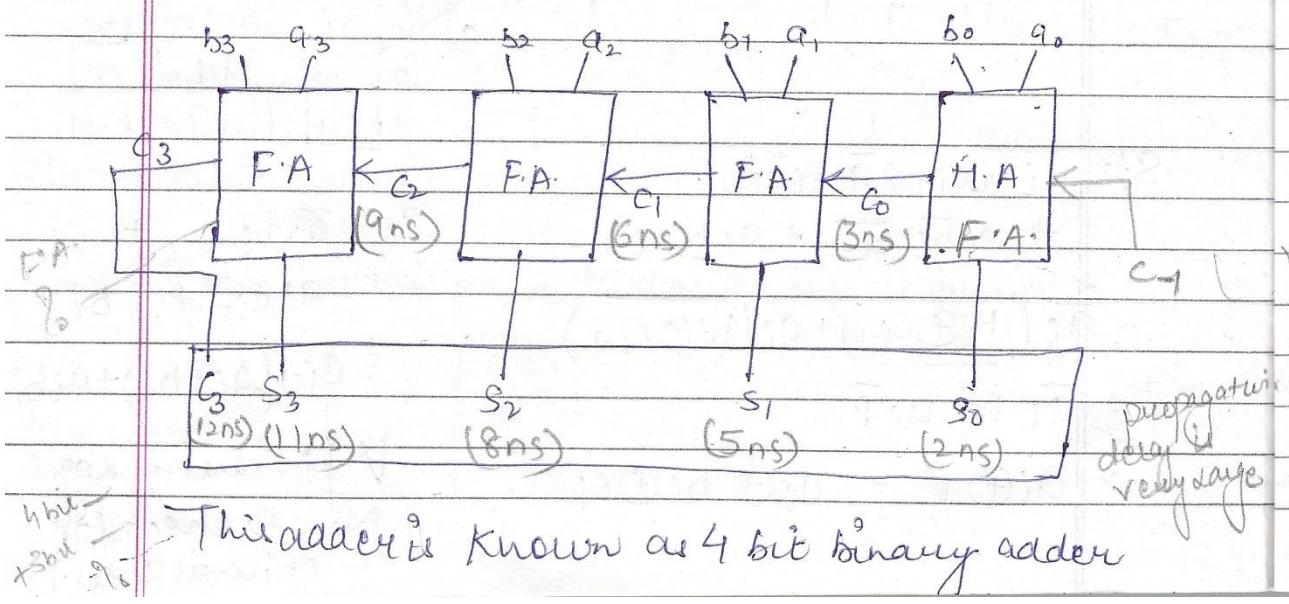
- * In case of two simultaneous O/Ps optimal soln. needs to be checked.



Full Adder

In full adder we are not implementing Minimum logic

MULTI BIT ADDITION



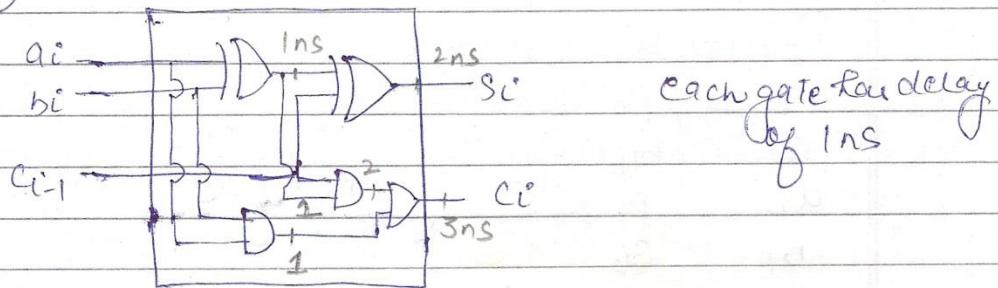
General Case - C_i

Mostly occurs in practical case
(in calculators)

Q.

$$\begin{array}{r} 5 \\ + 7 \\ \hline 12 \\ + 8 \\ \hline 20 \end{array}$$

but we do not use above ckt. b'z it has more propagation delay & hence less speed of calculation.



SPEED IMPROVEMENT OF ADDER

Carry Look Ahead Adder (CLA)

[.H/W is changed, cost is not affected much]

	a_i	b_i	C_{i-1}	C_i
Q.	0	0	0	0
delay	0	0	1	0
	0	1	0	0
Carry propagate state	0	1	1	1
	1	0	0	0
	1	0	1	1
Carry generate state	1	1	0	1
	1	1	1	1

Carry is generated by previous carry

Carry propagate state

$$P_i = a_i \oplus b_i$$

Carry propagate term

$$G_i = a_i \cdot b_i$$

Carry generate term

[uniquely represent]

$$C_i = a_i \cdot b_i + C_{i-1} (a_i \oplus b_i)$$

$$C_i = G_i + C_{i-1} P_i$$

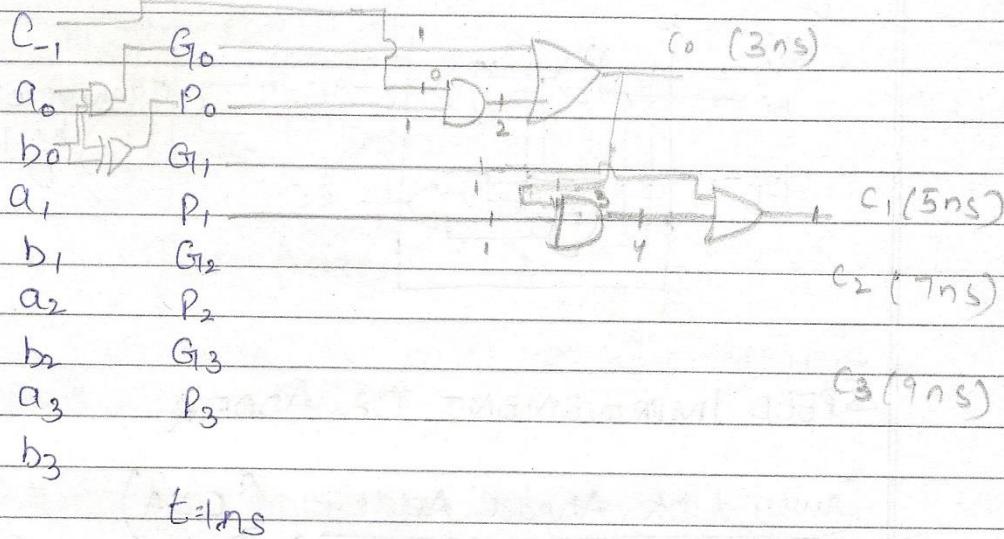
$$C_0 = G_0 + C_{-1} P_0$$

$$C_1 = G_1 + C_0 P_1$$

$$C_2 = G_2 + C_1 P_2$$

$$C_3 = G_3 + C_2 P_3$$

att: 0

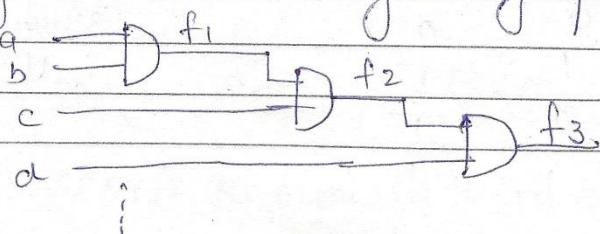


(Saving of only 3ns)

[Problem of Fan in, Fan out]

FAN IN, FAN OUT:

FAN IN - Max. No. of IP that can be given to a gate without degrading performance.

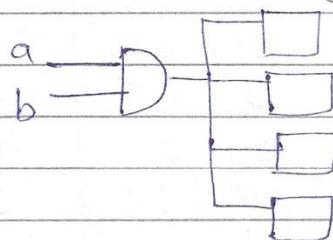


0 - 0v
1 - 5v

Tolerance

0 - 0.8v \rightarrow 0
4.2 - 5v \rightarrow 1

FAN OUT: Max. no. of off that can be taken from a gate without degrading performance.

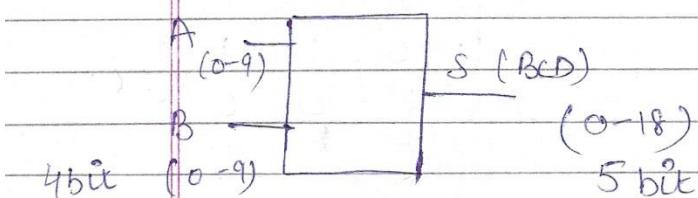


(performance becomes poor)

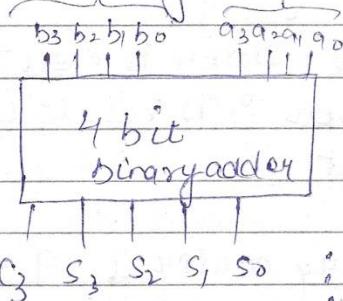
last gate
IP level
of all gates

Fan in, Fan out are figure of merit.
Other figure of merit are propagation delay,
power consumption, signal to noise ratio

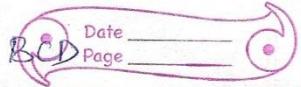
BCD Adder :-



Binary & BCD are same for 0-9. Then we
binary adder and after that convert it
into BCD using binary to BCD converter

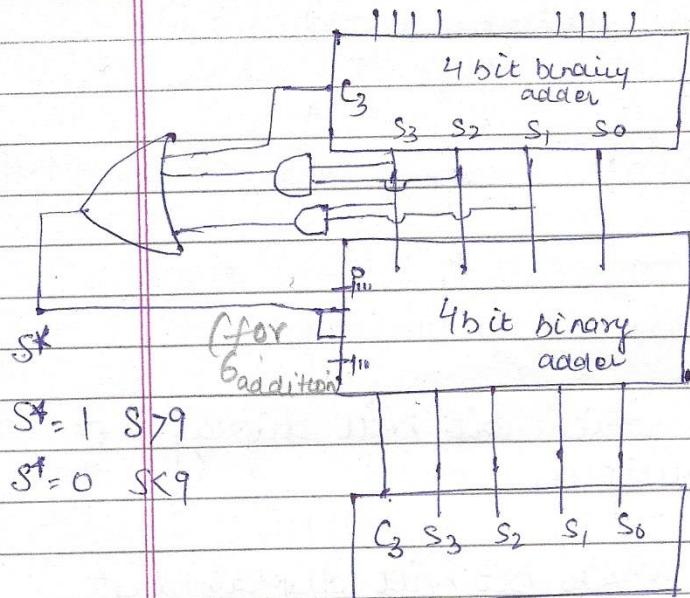


Binary



	C_3	S_3	S_2	S_1, S_0		C_3	S_3	S_2	S_1	S_0
0	0	0	0	0 0		0	0	0	0	0
1	0	0	0	0 1		0	0	0	0	1
2	0	0	0	1 0		0	0	0	1	0
3	0	0	0	1 1		0	0	0	1	1
4	0	0	1	0 0		0	0	1	0	0
5	0	0	1	0 1		0	0	1	0	1
6	0	0	1	1 0		0	0	1	1	0
7	0	0	1	1 1		0	0	1	1	1
8	0	0	1	0 0 0		0	1	0	0	0
9	0	0	1	0 0 1		0	1	0	0	1
10	0	0	1	0 1 0		1	0	0	0	0
11	0	0	1	0 1 1		1	0	0	0	1
12	0	1	1	0 0		1	0	0	1	0
13	0	1	1	0 1		1	0	0	1	1
14	0	1	1	1 0		1	0	1	0	0
15	0	1	1	1 1		1	0	1	0	1
16	1	0	0	0 0		1	0	1	1	0
17	1	0	0	0 1		1	0	1	1	1
18	1	0	0	1 0		1	1	0	0	0

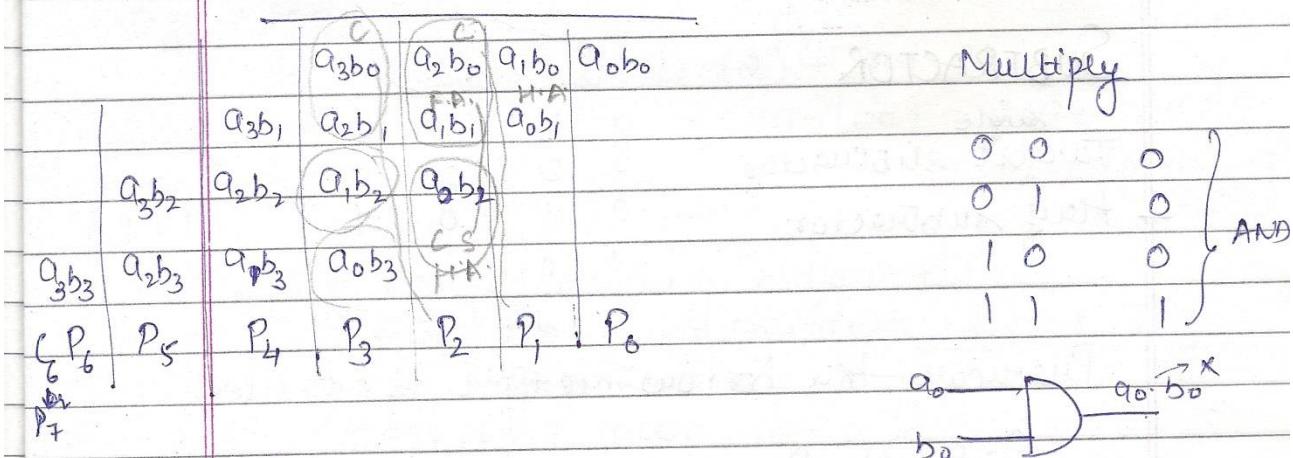
- 10 binary, 10 in binary is same as
- 10 in BCD is same as 16 in binary
upto 0-9 binary result = BCD result
- If result of binary adder is less than or equal to 9 then BCD result is same as binary result
- If result of binary adder is > 9 then add 6 in result of binary adder it will be result of BCD adder.



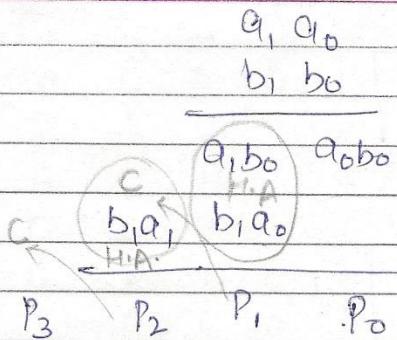
$13 \mid 6 \mid 13$ MULTIPLIER

$$A \rightarrow a_3, a_2, a_1, a_0$$

$$B \rightarrow b_3, b_2, b_1, b_0$$



By using AND gate and adder multiplication can be performed.



Pb

1) If $T_c > T_s$ then which bit will display last for given multipliers

2) If $T_s > T_c$ then which bit will display last for given multipliers

① $P_7 \quad 6T_c$

② $P_4 \quad 3T_c + 3T_s$

$$6T_c < 5T_c + T_s < 4T_c + 2T_s < 3T_c + 3T_s > 2T_c + 3T_s > T_c + 2T_s > T_s$$

$\cancel{+T_c + T_c + T_s}$

SUBTRACTOR

	a	b	Sub.	Borrow
single	0	0	0	0
Two bit subtractor	0	1	0	1
Half subtractor	1	0	1	0
	1	1	0	0

Practically this borrow method is not used

$$\begin{array}{r} a_3 \ a_2 \ a_1 \ a_0 \\ - b_3 \ b_2 \ b_1 \ b_0 \end{array}$$

but should know
the concept.

$$D = A - B$$

$$= A + (-B)$$

↪ 2's complement of B

In calculators addition and subtraction is done by same calc.

binary No. $a_{n-1} a_{n-2} \dots a_0 \cdot a_1 a_2 \dots a_m$

Base Radix (R) - It denotes total No. of symbols available to represent a No. in base R representation.

decimal form $\dots + a_1 \cdot 2^1 + a_0 \cdot 2^0 + a_1 \cdot 2^{-1} + a_2 \cdot 2^{-2} + \dots$

COMPLIMENT

$$N_{10} = 1234$$

$$10's \text{ complement of } N_{10} = (10)^4 - 1234$$

$$N_x = (a_n a_{n-1} \dots a_1) \quad \text{No. of digits}$$

$$\rightarrow 2's \text{ complement of } N_x = (2)^n - N_x$$

$$\text{e.g. } 1001$$

$$2's \text{ complement of } N_x = (2)^4 - 1001 \\ = 16 - 1001 = 10000 - 1001$$

$$10000$$

$$1001$$

$$0111$$

$$= 0111 \quad [100000 \text{ rega.}]$$

$$\rightarrow (2)^n = \text{largest No. using } n \text{ digits} + 1$$

= Smallest No. using $(n+1)$ digits in base using n representation

$$9999 + 1 = 10000$$

$$\rightarrow 2^n - 1 = \text{largest No. using } n \text{ digits}$$

$$9's \text{ complement of } N_{10} = ((10)^4 - 1) - 1234 = 1$$

$$9999 - 1234 = 8765$$

No borrow is rega.

1's complement of 1001 = $(2^4 - 1) - 1001$
 $= 1111 - 1001 = 0110$

∴ 2's comp. of 1001 = 1's comp. + 1
 $= 0110 + 1 = 0111$

* γ 's comp. of $N_x = (\gamma - 1)$'s comp. of $N_x + 1$

* $(\gamma - 1)$'s comp. of $N_x = \underbrace{(\gamma^n - 1)}_{\text{Largest no. using } n \text{ digits}} - N_x$

- Bit by bit comp. of a no. is 1's complement
- Digit by digit comp. of a no. is $(\gamma - 1)$'s complement

2's comp. representation 4 BIT SIGN MAGNITUDE REPRESENTATION OF BINARY NUMBER.

Sign bit mag. bit
MSB LSB

+0 0 0 0 0

+1 0 0 0 1

+2 0 0 1 0

+3 0 0 1 1

+4 0 1 0 0

+5 0 1 0 1

+6 0 1 1 0

+7 0 1 1 1

-8 1 0 0 0

-7 1 0 0 1 2's comp. of +7

-6 1 0 1 0

-5 1 0 1 1

-4 1 1 0 0

-3 1 1 0 1

-2 1 1 1 0

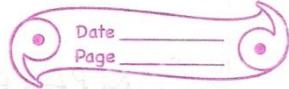
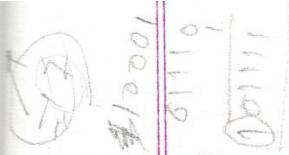
-1 1 1 1 1

[Range]

tve no. to to $(2^{n-1} - 1)$

-ve no. -1 to $-(2^{n-1})$

range +0 to +7 & -1 to -8



→ In 1's complement $+0 = 0000$ $-0 = 1111$
 But since $+0 = -0$ Hence 1's comp is not used
 Moreover its range is
 $+0$ to $+7$ [and -0 to -7] [4 bit]

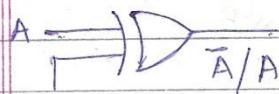
i.e. $\{+0 \text{ to } [2^{n-1} - 1]\}$ and $-0 \text{ to } -(2^{n-1} - 1)$ [n bit]

→ ACTUAL SUBTRACTION

$$D = A + (-B) \quad -B \rightarrow 2^{\text{th}} \text{ complement of } B$$

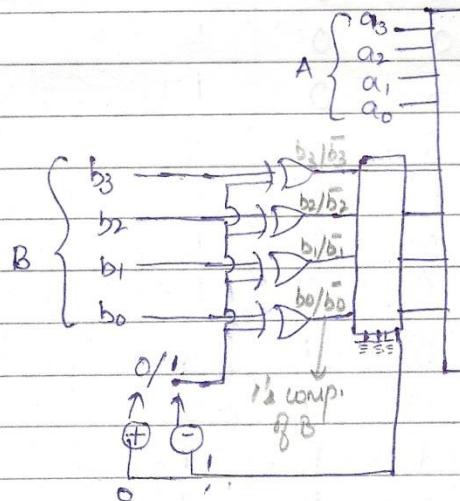
D = A + 2^{\text{th}} \text{ comp. of } B rega. ctet.

Adder Cum Subtractor

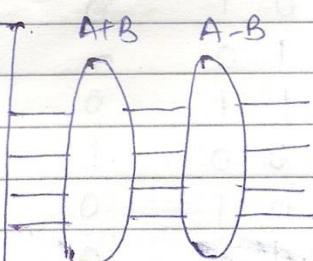


XOR 1/0

0 0 0
0 1 1
1 0 1
1 1 0



A+B A-B



5 digits in
result means
signed no

$$\begin{array}{r} 1100 \\ - 0001 \\ \hline \end{array} = \boxed{-11} \quad \begin{array}{r} 12-1 \\ - 1010 \\ \hline \end{array} = \boxed{-9} \quad \begin{array}{r} 0001 \\ - 110 \\ \hline \end{array} = \boxed{-9} \quad \begin{array}{r} 0010 \\ - 0001 \\ \hline \end{array} = \boxed{1}$$

$$\begin{array}{r} 1110 \\ + 1 \\ \hline 1111 \\ + 1100 \\ \hline 10011 = \boxed{-11} \end{array} \quad \begin{array}{r} 0101 \\ + 0110 \\ \hline 0111 = \boxed{7} \end{array} \quad \begin{array}{r} 0110 \\ + 0010 \\ \hline 10001 = \boxed{-1} \end{array}$$

Overflow error

COMPARATOR

for a, b single bit no.

a	b	F_1 $A > B$	F_2 $A < B$	F_3 $A = B$	F_4 $A \geq B$	F_5 $A \leq B$
0	0	0	1	1	1	1
0	1	0	1	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	1	1

$\frac{\downarrow}{\bar{a}b}$ \overline{ab}
 $\bar{a}+\bar{b}$ $\bar{a}+b$

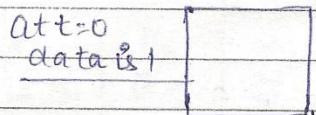
$A \rightarrow a, a_0$

$B \rightarrow b, b_0$

A a, a ₀	B b, b ₀	F_1 $A > B$	F_2 $A < B$	F_3 $A = B$	F_4 $A \geq B$	F_5 $A \leq B$
0 0	0 0	0	0	1	1	1
0 0	0 1	0	1	0	0	1
0 0	1 0	0	1	0	0	1
0 0	1 1	0	1	0	0	1
0 1	0 0	1	0	0	1	0
0 1	0 1	0	0	1	1	1
0 1	1 0	0	1	0	0	1
0 1	1 1	0	1	0	0	1
1 0	0 0	1	0	0	1	0
1 0	0 1	1	0	0	1	0
1 0	1 0	0	0	1	1	1
1 0	1 1	0	1	0	1	1
1 1	0 0	1	0	0	1	0
1 1	0 1	1	0	0	1	0
1 1	1 0	1	0	0	1	0
1 1	1 1	0	0	1	1	1

SEQUENTIAL CIRCUITS

Date 14/6/2013
Page



at $t=4\text{ns}$ data is reqd. ?

then storage is needed

- for this we may use $t=0, D_0^0, D_0^1, D_0^0, D_0^1$ at $t=4\text{ns}$

This is basically delay not storage because after 4ns data is not available

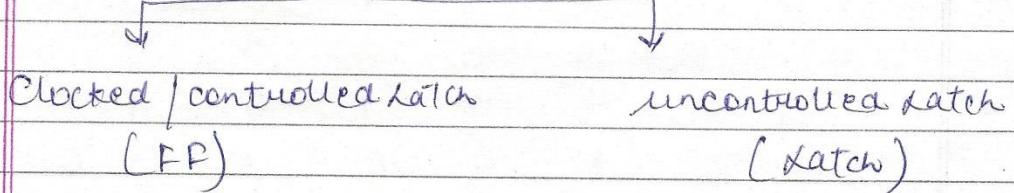
- Stored data must be retrieved at any instant
- Storage + retrieval at any instant = memory

SINGLE BIT STORAGE ELEMENT

It is basic building block for sequential circuit - known as latch

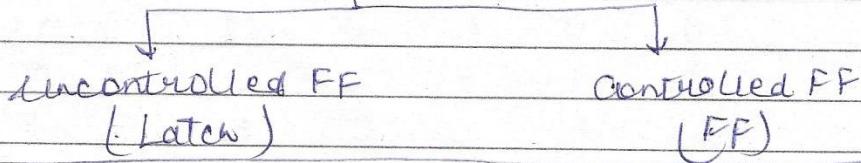
better ✓

latch



FF is advance version of latch

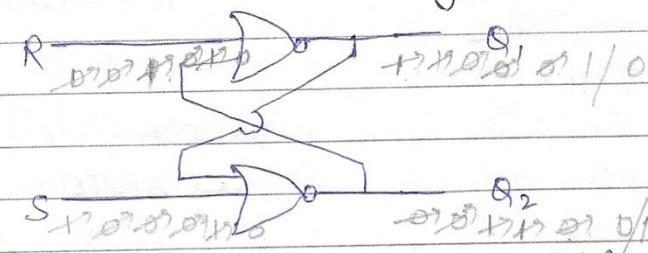
FF



latch / uncontrolled FF

FF / controlled latch

(1.) S-R latch with NOR gate



NOR gate

0 0 1

0 1 0

1 0 0

1 1 0

Next O/P depends on present O/P

$\left\{ \begin{array}{l} \text{if I/P is 1 then} \\ \text{O/P is zero} \end{array} \right.$

S R Q_1 Q_2

1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1

Storage

+ control

same I/P same O/P

00 I/P data is stored

$Q_2 = Q_1$) I/P can be

maintained

in memory

electrons

depend on which gate is faster

\therefore unreliable system

Q_2 is inverted form of Q_1

Information available at S will be stored at Q_1

Q_2 do not contain independent information

\therefore 1 bit storage

drawbacks

- restricted state $R=1, S=1$

- Storage is 1 bit but I/P is two terminal [Not available]

- Storage & maintenance are from same terminal

\therefore data may change due to change in O/P of ctrl (adder)

Characteristic table is diff. from T-T

Date: _____
Page: 59

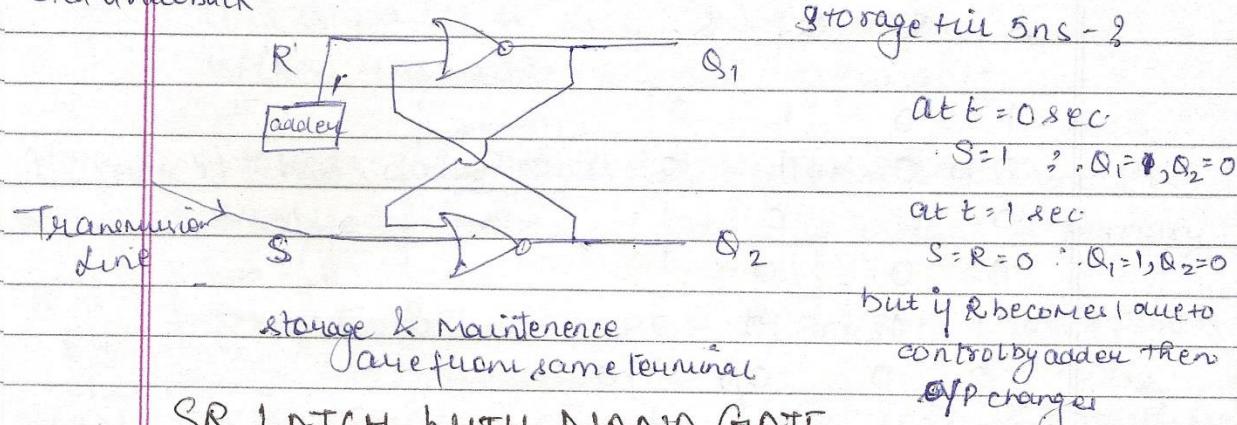
Characteristic table

S - Set (1)

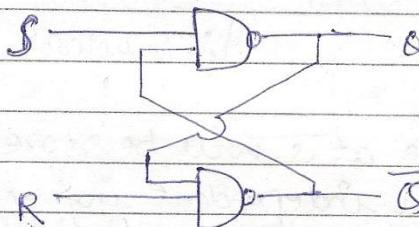
R - Reset (0)

S	R	Q_{n+1}	\bar{Q}_{n+1}	
0	0	Q_n	\bar{Q}_n	memory state
Reset I/P	0	1	0	reset state
Set I/P	1	0	1	set state
	1	1		Restricted state different from don't care

3rd drawback



SR LATCH WITH NAND GATE



NAND gate

0 0 1

0 1 1

1 0 1

1 1 0

[1HPO, OP is 1]

S	R	Q_{n+1}	\bar{Q}_{n+1}
0	0	R.S.	

Reset I/P 0 1 1 0 Set state

Set I/P 1 0 0 1 Reset state

$Q_1 \quad Q_1 \quad Q_n \quad \bar{Q}_n$ Memory state

SR Latch with NOR gate & NAND gate are same basically

JFP xnor The drawbacks are same as that of SR Latch with NOR gate.

- JFP should be free from maintenance i.e. extra control terminal is required to maintain data

SR

data available at R is stored at Q_{n+1}

RS

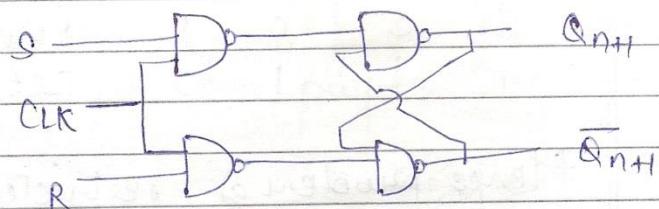
data available at S is stored at Q_{n+1}

(use depend on application)

- [Not to remember Naming & T.T.]

CONTROLLED LATCH (FF)

(1) SR Flip Flop



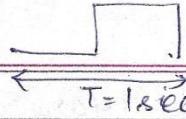
CLR	S	R	Q_{n+1}	\bar{Q}_{n+1}	
0	X	X	Q_n	\bar{Q}_n	Memory state
1	0	0	Q_n	\bar{Q}_n	
0	1	0	1	0	R-S
1	1	0	1	0	Reset
1	1	1	0	1	Set
1	1	1			Restricted

[JFP is 1
O/P is inverted]

- Drawback of controlling is removed but other 2 exist

- CLK is made up of crystal quartz. It is a high freq. device and becomes unstable at low freq.

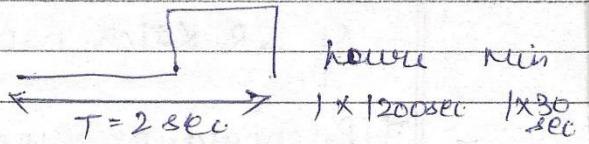
High freq. settling period



Date _____
Page _____
Settling time min.

1x3600sec 1x60sec.

Low freq. large settle period

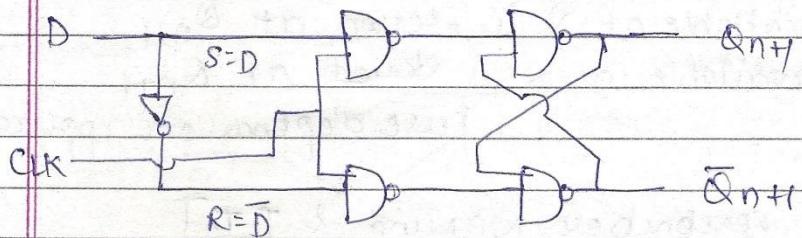


hours min

1x1200sec 1x30sec

i.e. freq. division decrease, less calculation, cheap H/w
(watcher)

(2) ~~data~~ D FLIP FLOP



CK D Q_{n+1} Q̄_{n+1}

0 X Q_n Q̄_n M.S.

1 0 0 1 Reset state

1 1 1 0 Set state

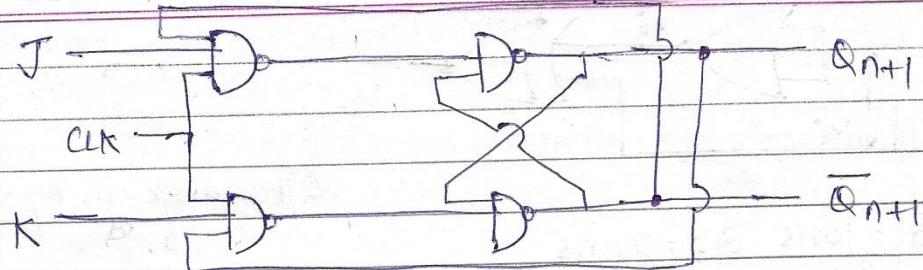
Hence problem of restricted state 11 is removed
& it is used to store single bit data using
1 terminal

Since all drawbacks are removed Hence it is used
practically.

Now problem is to convert restricted state to useful
form

(3) J-K FLIP FLOP

3 I/P NAND gate is used



Assume at $t=0$ $\begin{matrix} \text{CLK} & \text{J} & \text{K} & \text{Q}_{n+1} & \bar{\text{Q}}_{n+1} \\ 1 & 1 & 1 & 0 & 1 \\ \end{matrix}$

$\begin{matrix} 1 & 1 & 1 & \xrightarrow{0} & 1 \rightarrow 0 \rightarrow 1 \\ & & & \downarrow & \\ & & & 0 & \dots \end{matrix}$

→ Uncontrolling toggling - racing

→ Transition before ~~the~~ ie. before becoming 1 it goes to 0 and before becoming 0 it goes to 1
ie. No proper settling, No proper 5th v is available at 1 and No proper 0th v is available at 0

Toggle?

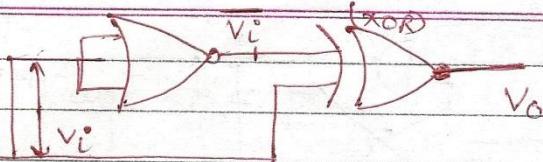
CLK - IMMMI
 $J=1, K=1$

Ques:

Rainy day

Date 15/6/18
Page

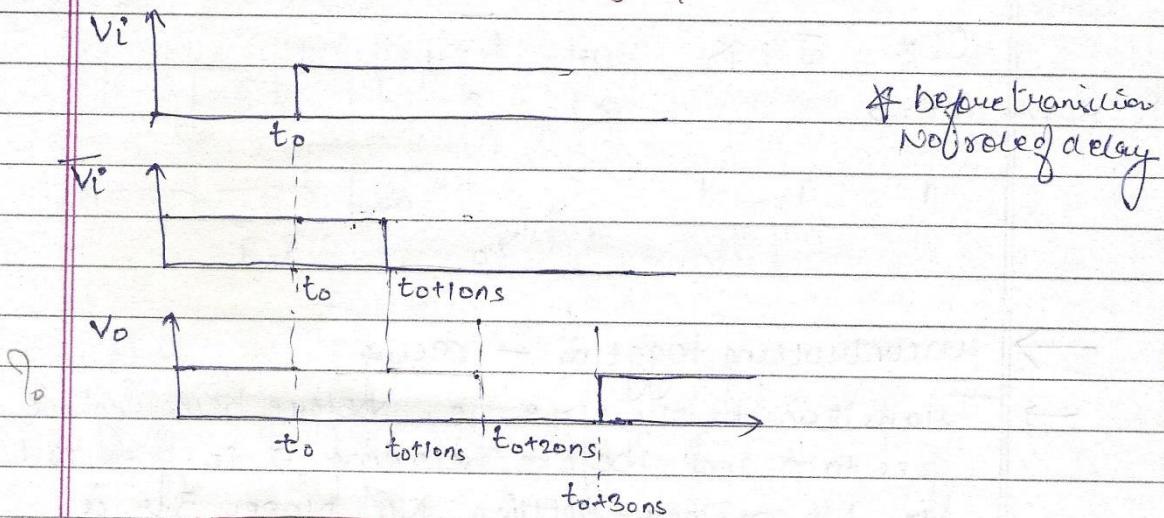
e.g.



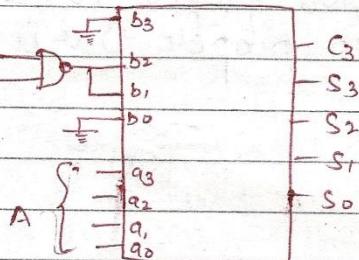
$$G_1 = 10\text{ns} \quad G_2 = 20\text{ns}$$

V_i makes abrupt change from 0 to 1 at $t=t_0$

O/P waveform of V_o ?



e.g.



A is 4 bit binary No. $A > 9$

then S(O/P) is

(i) 2's complement of A

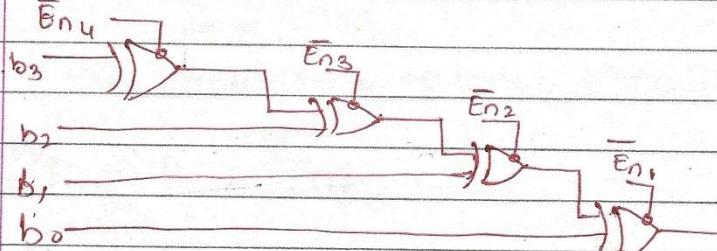
(ii) 1's comp. of A

(iii) S is BCD code of A

(iv) all of these.

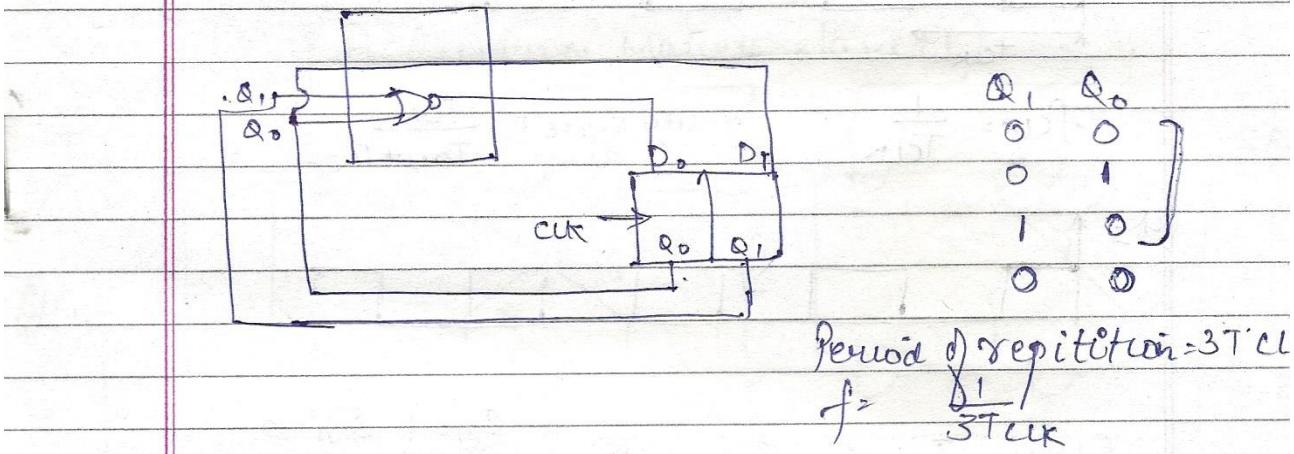
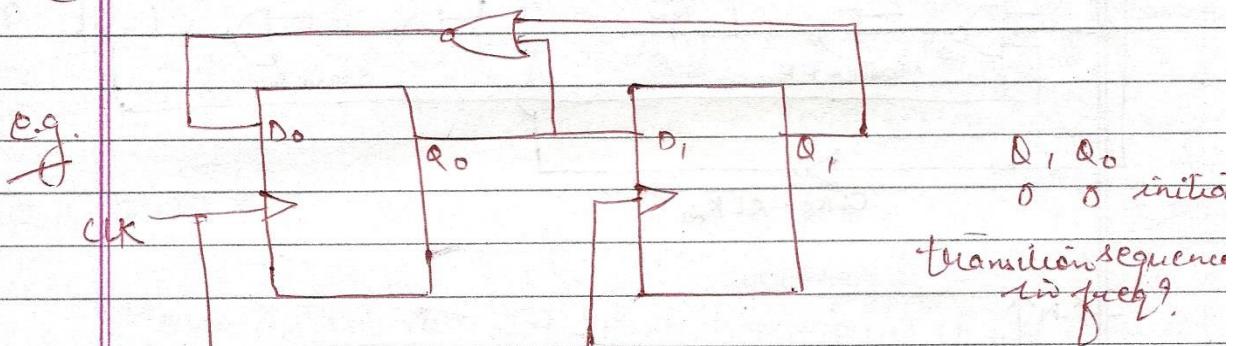
Since $A > 9$ \therefore S is BCD code of A

e.g.



?

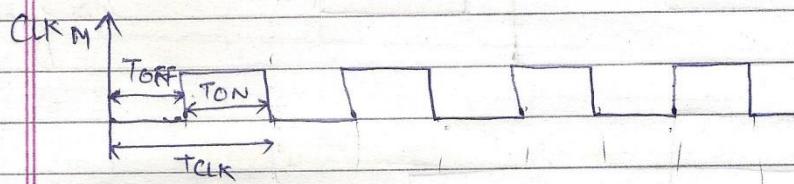
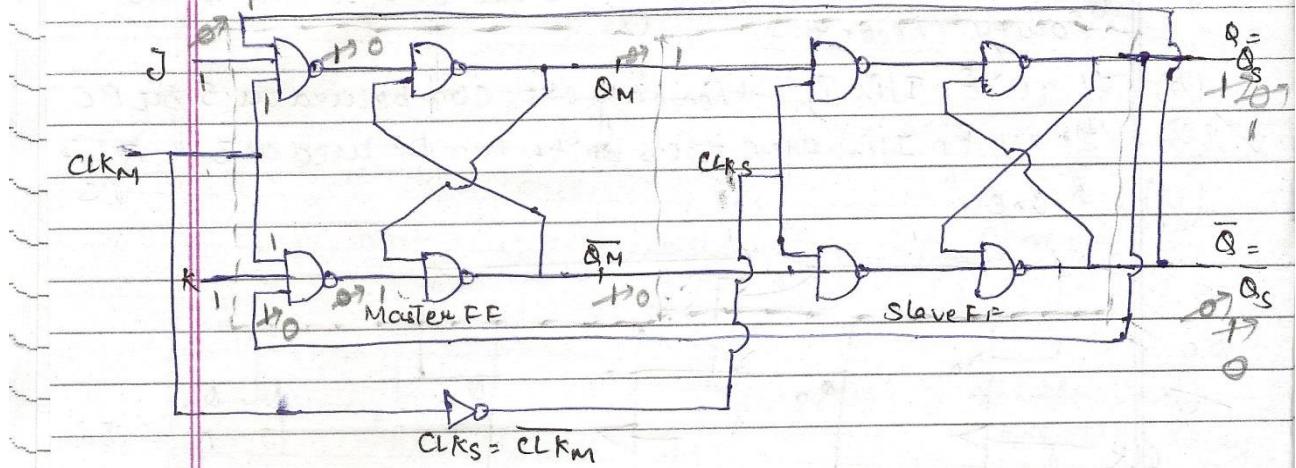
- ✓ (a) If $E_{n-4}, E_{n-3}, E_{n-2}$ are zero then given ckt. can be used as 4 bit Parity checker
- (b) If all $E_n I/Ps$ is 1 then only ckt. can be used as 5 bit PC
- ✓ (c) If all $E_n I/Ps$ are 0 then ckt. can be used as 3, 4, 5 bit PC
- (d) None



- * For 2 variables use Truth Table
- * For more than 2 variables use K-map

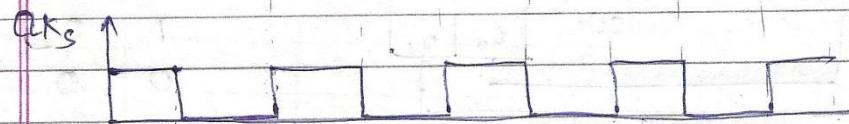
Date 17/6/2013
Page

MASTER SLAVE RS PULSE FLOP



$$f_{CLK} = \frac{1}{T_{CLK}}$$

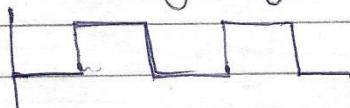
$$\text{Duty cycle} = \frac{T_{ON}}{T_{ON} + T_{OFF}}$$



$Q_n = 1 \quad \bar{Q}_n = 0$
 $Q_{n+1} \quad \bar{Q}_{n+1}$ [Continued toggle mode]

$\therefore 1$ transition occurs after 1 CLK Period

proper settled signal of 5V or 4.8V is obtained



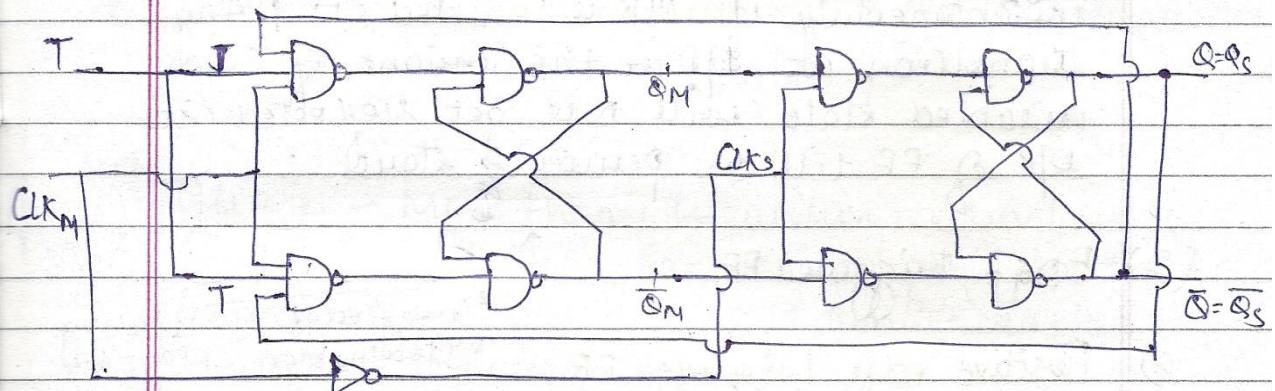
Characteristic Table

CLKM	J	K	Q_{n+1}	\bar{Q}_{n+1}
0	x	x	0	Memory state
1	0	0	Memory state	removal of 3 states do not effect behaviour of FF
1	0	1	0	
1	1	0	1	
1	1	1	\bar{Q}_n	Q_n . Toggle mode controlled

$CLK_M = J = K = 1$ can be used for storage of 0 bit, 1 bit & even toggling
 \downarrow
 $CLK_M \neq 0$

Hence 3 states do not effect behaviour of FF if removed

T FLIP FLOP (from Master & slave FF)



CLK_M T Q_{n+1} \bar{Q}_{n+1}

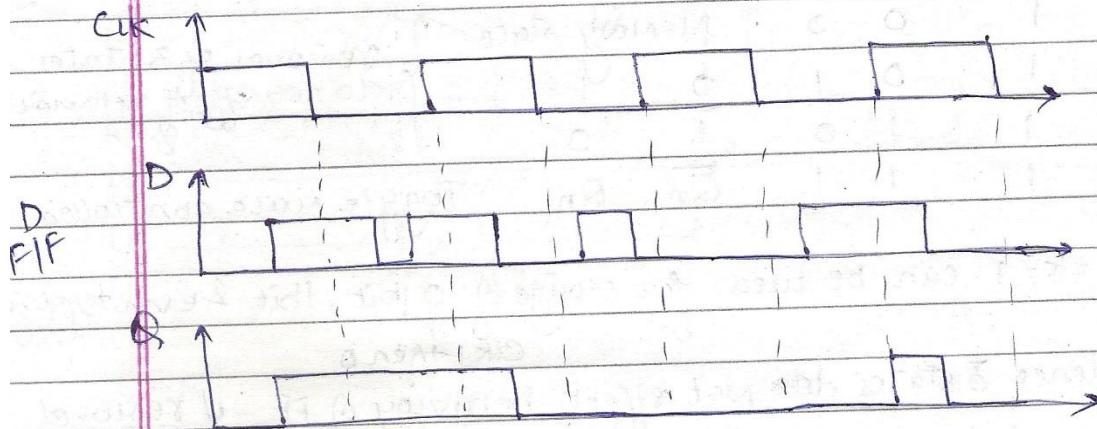
0	x	M.S.
1	0	M.S.
1	1	Q_n \bar{Q}_n toggle controlled

T flip flop from Master Slave FF is useful but
 $T = " "$ $J = K = " "$ "NOT".

Hence useful FFs are D, Master slave, T with master slave

TRIGGERING MECHANISM OF FF

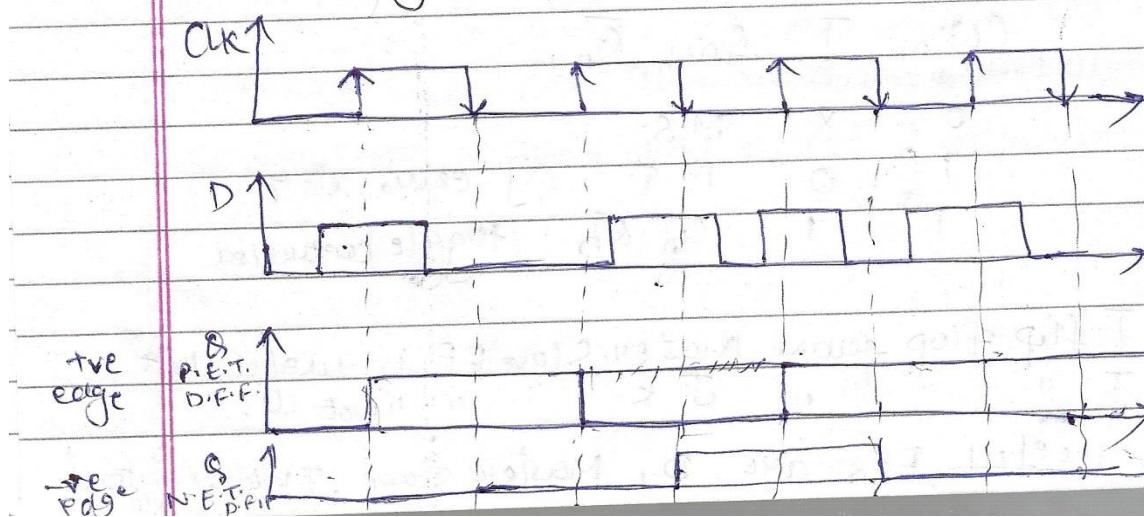
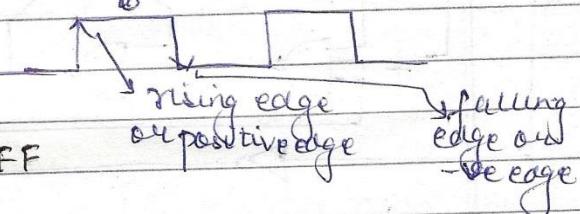
(1) Level triggering FF



Any transition at I/P will get reflected at O/P of FF instantaneously if Clk is enabled (1). Any transition at I/P at the instant of clock disabled state will not get reflected into O/P of FF till ON period of state.

(2) EDGE triggered FF

a) Positive edge triggered FF



Positive edge triggered P/F - F/F will be sensitive to positive edge of CLK. Any transition at I/P will get reflected into O/P only at the instant of +ve edge. O/P will remain in that state at least for 1 CLK period. ~~hence~~ in case of -ve.

b) Negative edge triggered FF :

F/F will be sensitive to -ve edge of CLK. Any transition at I/P will get reflected into O/P only at the instant of -ve edge. O/P will remain in that state at least for 1 CLK period.

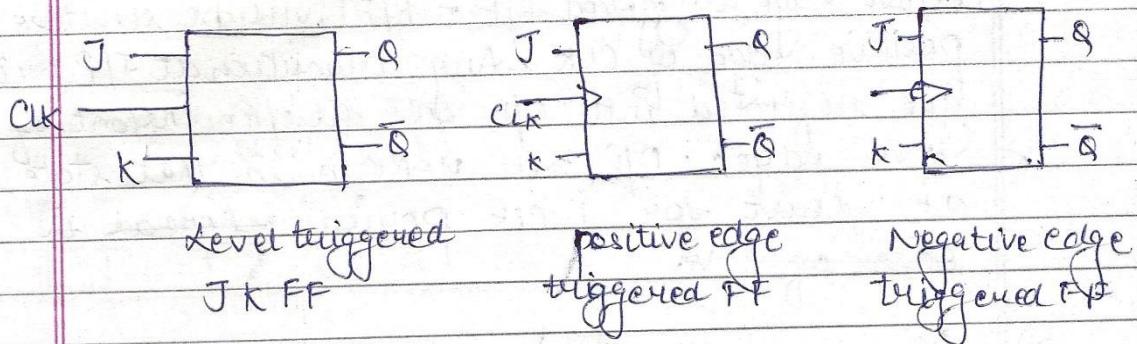
* More than 1 CLK period can exist b/w two transition at O/P at not less than 1 CLK period.

[#]
(level)
trigg.) Glitches - More than 1 transition within 1 clock period. i.e. signal has not properly settled down. Hence Glitches are undesirable and are ignored by slave.

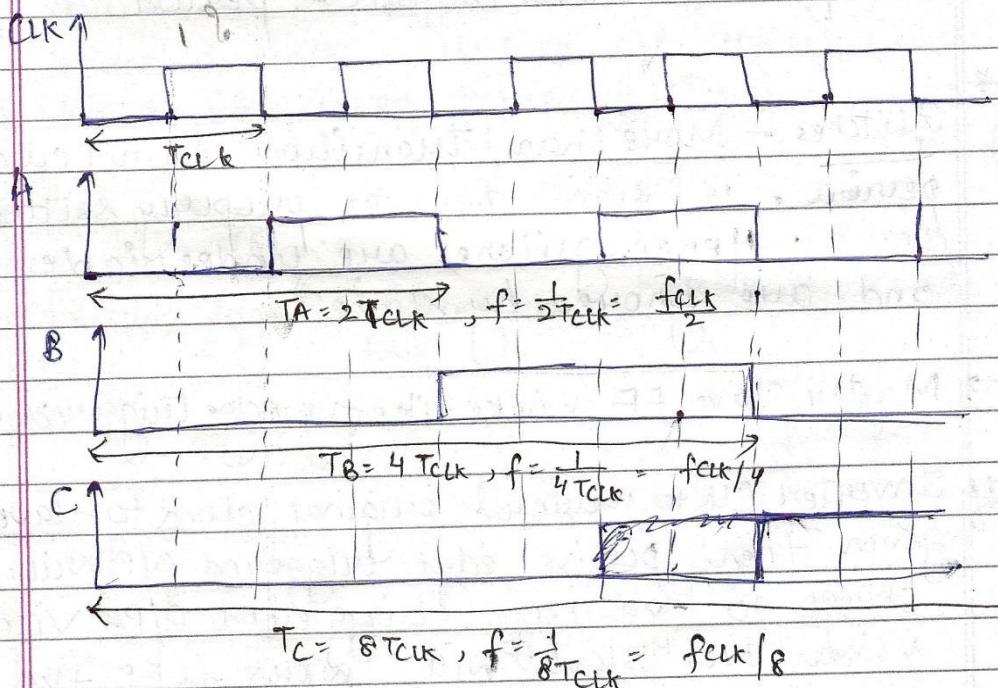
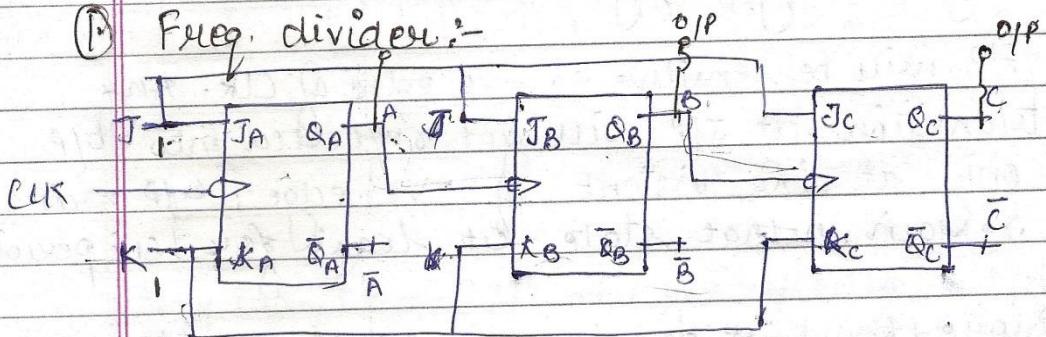
→ Master Slave FF works like -ve edge triggered FF

→ If Inverted CLK to master & original clock to slave is given then positive edge triggered O/P will change to -ve edge triggered O/P & vice versa and M'slave FF will work like +ve edge triggered FF.

APPLICATION OF FF :-



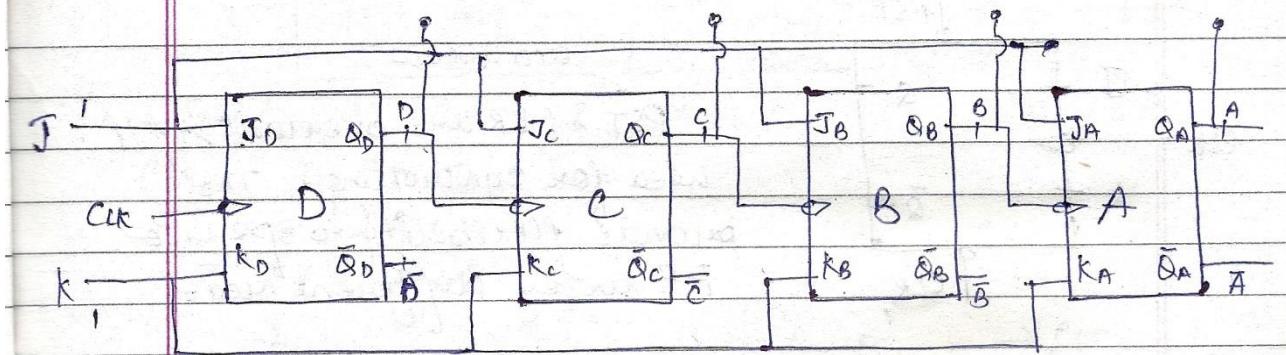
(B) Freq. divider :-



* frequency of nth O/P = $f_{clk}/2^n$

	A	B	C	
Clk Period	0	0	0	Reset State
1	1	0	0	
2	0	1	0	
3	1	1	0	
4	0	0	1	
	0	1		
		1		
			1	

COUNTER:



Clk Period A B C D

	A	B	C	D	
Mod16 Counter	0	0	0	0	Counting of clk signals upto 15
	1	0	0	1	Next 0
	2	0	0	1	∴ Count range 0-15
	3	0	0	1	Total length of count = 16
	4	0	1	0	
	5	0	1	0	Asynchronous Counter
	6	0	1	1	Up counter
	7	0	1	1	Up counter
	8	0	0	0	(Modulo) Counter
	9	1	0	0	Mod Modulus (length)
	10	1	0	1	
	11	1	0	1	
	12	1	1	0	
	13	1	1	0	

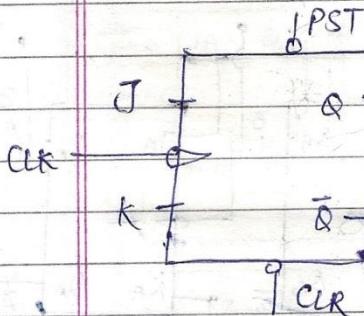
* Four down counters taken O/P from $\bar{A}, \bar{B}, \bar{C}, \bar{D}$

→ Synchronous Counter is the counter where all basic blocks operate in synchronism with each other.

→ $0 \rightarrow 2^n - 1$ MOD counter

$0-5$ counter

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$



PST & CLR are overwriting J/K
used for controlling. These
provide flexibility to operate
FF under different mode.

* PST is active independent of J, K, CLR (O/P is 1)

* CLR is " " " " " " " " " " " 0

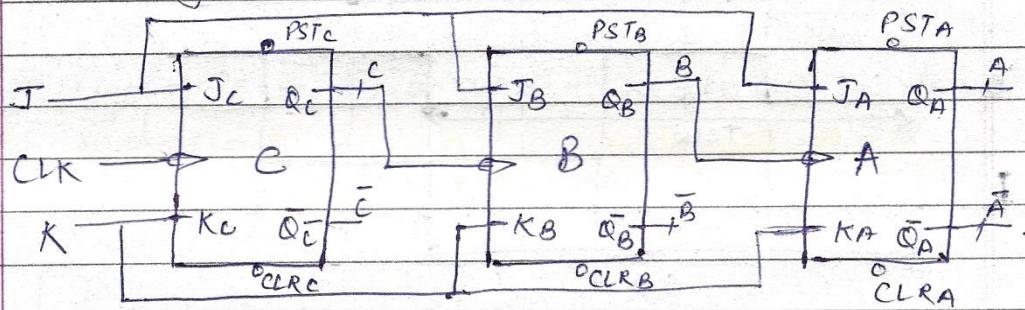
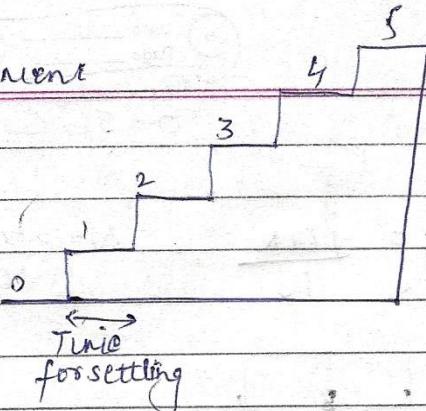
CLK	J	K	PST	CLR	
X	X	X	0	0	Restated state
X	X	X	0	1	1
X	X	X	1	0	0
X	X	X	1	1	Normal operation of FF (dependent on J, K, CLK)

Pb

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

requirement

Date _____
Page _____



CLK Period A B C

0 0 0 0

1 0 0 1

2 0 1 0

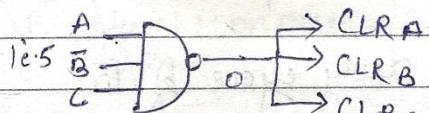
3 0 1 1

4 1 0 0

5 1 0 1

6 1 1 0

7 1 1 1



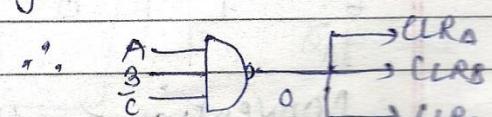
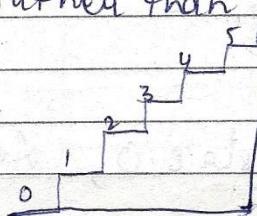
O/P 0

and it will shift to
000

glitch

But problem is at 5 is reached within very small time it makes O/P 0. But we require 5 for some time. Hence if 0-5 is reqd. we give the condition of 6 to make complete i.e. rather than at 5 glitch will occur at 6

glitch at 1 step
ahead has no
problem



$$T_{CLK} = 0 - 5 = 6$$

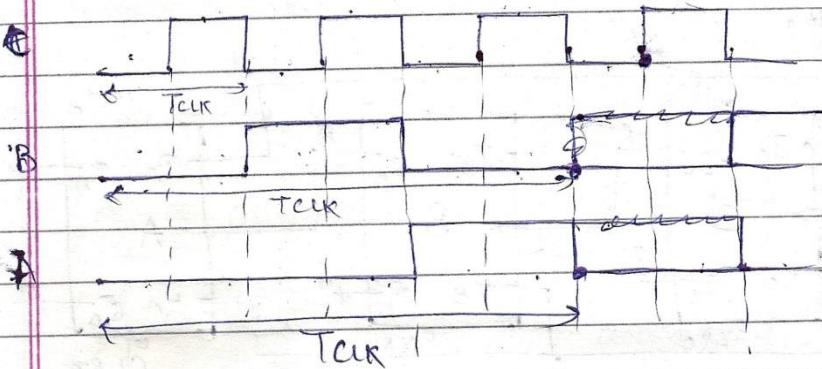
$$f_{CLK} = \frac{1}{T_{CLK}} = \frac{1}{6} f_{CK}$$



MOD-N counter

$$f_{OUT} = f_{CLK} \text{ of device} \quad N$$

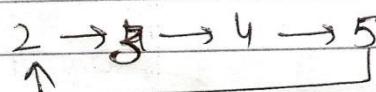
$N = \text{desired count}$



$$f_{CLK_A} = \frac{f_{CK}}{6}, \quad f_{CLK_B} = \frac{f_{CK}}{6}, \quad f_{CLK_C} = \frac{f_{CK}}{2}$$

$$\text{i.e. } f_{CLK_A} = f_{CLK_B} = f$$

$$\rightarrow \text{duty cycle of } B = \frac{2}{6} = \frac{1}{3}$$

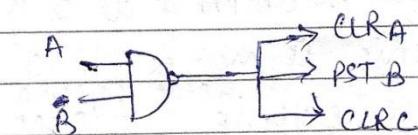


alternate

Clk Period A B C

Modulo 4 counter

0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0



Nonclock
transition

Convention - initial state 0, level triggered

Since initially system is at 0, it will take sometime to follow sequence.

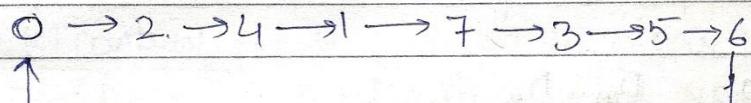
Q. What will be count of this counter at the instant of 13 clock pulses
5

9. Settling time - time reqd. to settle device for normal operation.
6 T_{CK} - latency

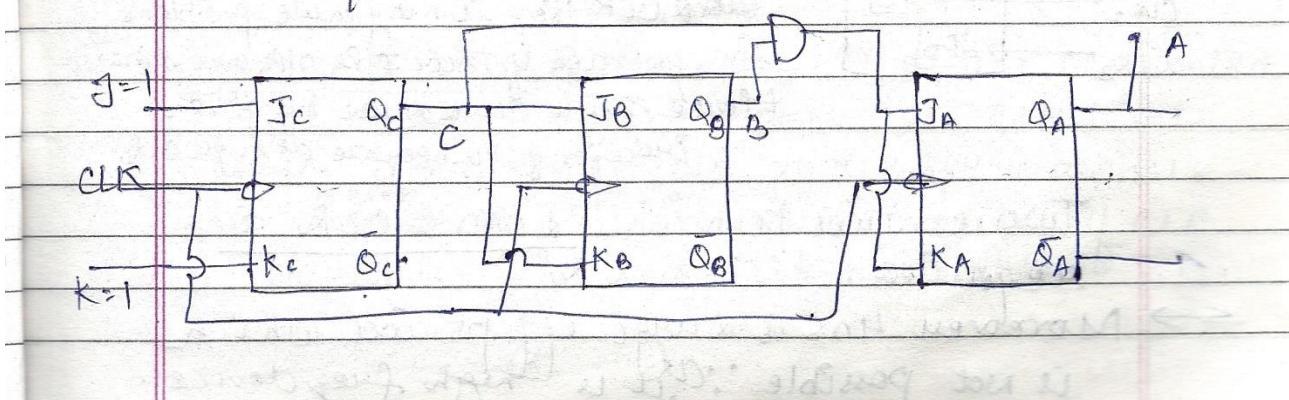
→ Problem is propagation delay
Propagation delay of asynchronous counter = $n T_{CK}$ n = No. of flip flops

18/6/B

To overcome this we use synchronous counters



complexity of asynchronous counters in case of random variable count increases & power consumption also increases.



CLK A B C

0	0	0	$2^{MS} \downarrow T$
1	0	0	2^T
2	0	1	2^T
3	0	1	$2^{MS} \downarrow T$
4	1	0	2^T
5	1	0	2^T
6	1	1	2^T
7	1	1	2^T

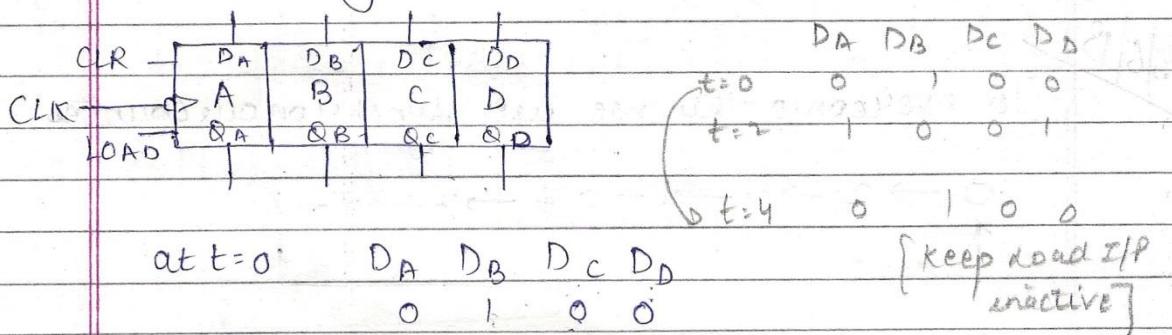
- In syn. counter,
propagation delay is
single T CLK

- FAN IN, FANOUT problem
as O/P signal is degraded

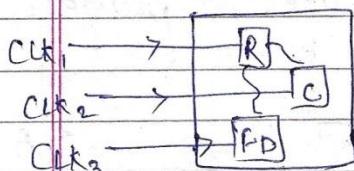
REGISTER (multibit storage device)

best FF for storage - D Flip flop.

4 bit storage



data is regd. at t=4 ns



The different seq. ckt's of given diff. ~~clock~~ CLK then it may cause problems ^(due to time difference) in operation if their I/Ps O/Ps are interlocked.
Hence single clock must be given.
Processor of CPU operate at single CLK.

Two control terminals LOAD & CLR are required.

→ Moreover 4ns is large off period which is not possible ; it is high freq device.

The same CLK may be used for activation of other devices. But if we keep it 0 to store the data then the O/P of other device is also zero inspite of fact that it may be reqd.

Load Control - It is used to load the data available at I/P terminals of a register at O/P terminal of resistor.

Two modes of loading of data are:-

- (i) Synchronous Loading (ii) Asynchronous Loading

→ Synchronous Loading - It means load control will operate in synchronism with CLK. That means data available at I/P terminal will be stored at O/P terminal of register if load control is high and CLK is enabled.

→ Asynchronous loading - It means load control will operate independent of CLK, i.e. as soon as load pulse arrives data available at I/P terminal will be stored in register independent of CLK.

CLR control - It is used to clear the data available at O/P terminals of register.

CLR control can also be operated in two modes:-

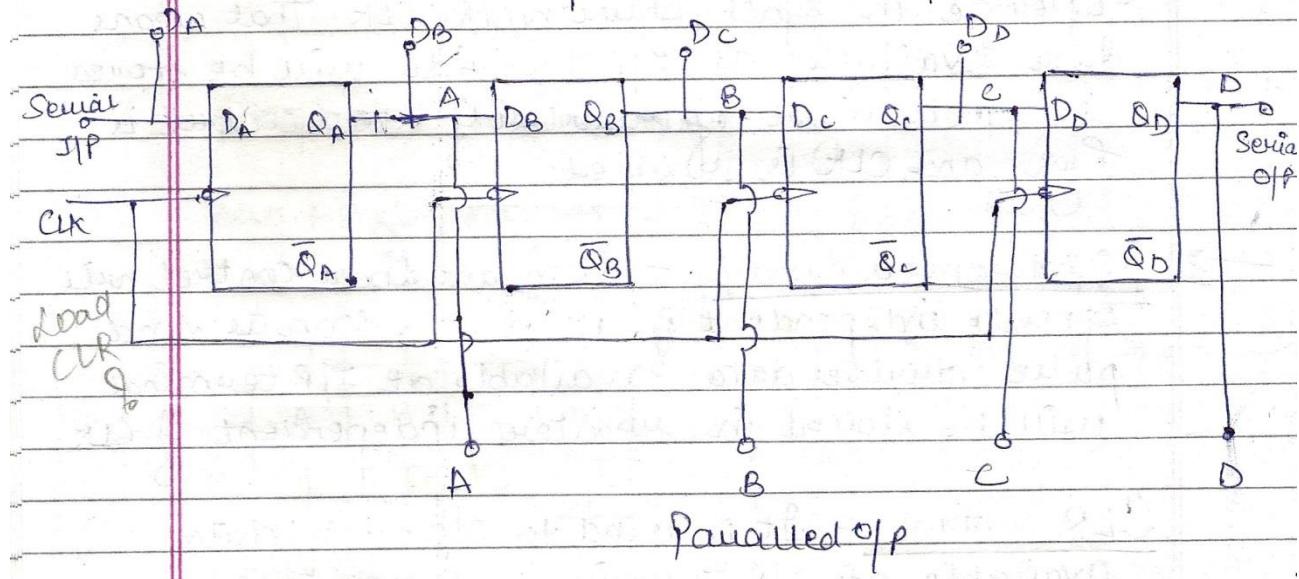
- (i) Sync mode of clearing (ii) Asynchronous mode of clearing

→ Synchronous Mode of clearing - It means CLR control will operate in synchronism with CLK. That means data available at I/P terminal will be stored at O/P terminal of register if CLR control is high and CLK is enabled.

→ Asynchronous Mode of clearing :- It means CLR control will operate independent of CLK i.e. as soon as CLR pulse arrives data available at I/P terminal will be reset independent of CLK.

SHIFT REGISTERS :-

It can be operated in Multiple Modes.



Serial I/P	A	B	C	D
at t=0 assume	IN ₁	1	0	0
	IN ₂	IN ₁	1	0
	IN ₃	IN ₂	IN ₁	0
	IN ₄	IN ₃	IN ₂	IN ₁
		IN ₄	IN ₃	IN ₂

→ data shifts ∴ it is called shift register.

→ data I/P at 1st will be O/P at 1st
IN₁ I/P I_E, O/P IN₁ I_E

4 Modes of operation of Shift Register :-

- 1) SISO Serial I/P serial O/P - delay
- 2) SIPO Serial I/P parallel O/P - Modem [PC connected to transmission line]
- 3) PISO Parallel I/P serial O/P - encols.
- 4) PIPO Parallel I/P parallel O/P - storage

∴ Multiple modes operation

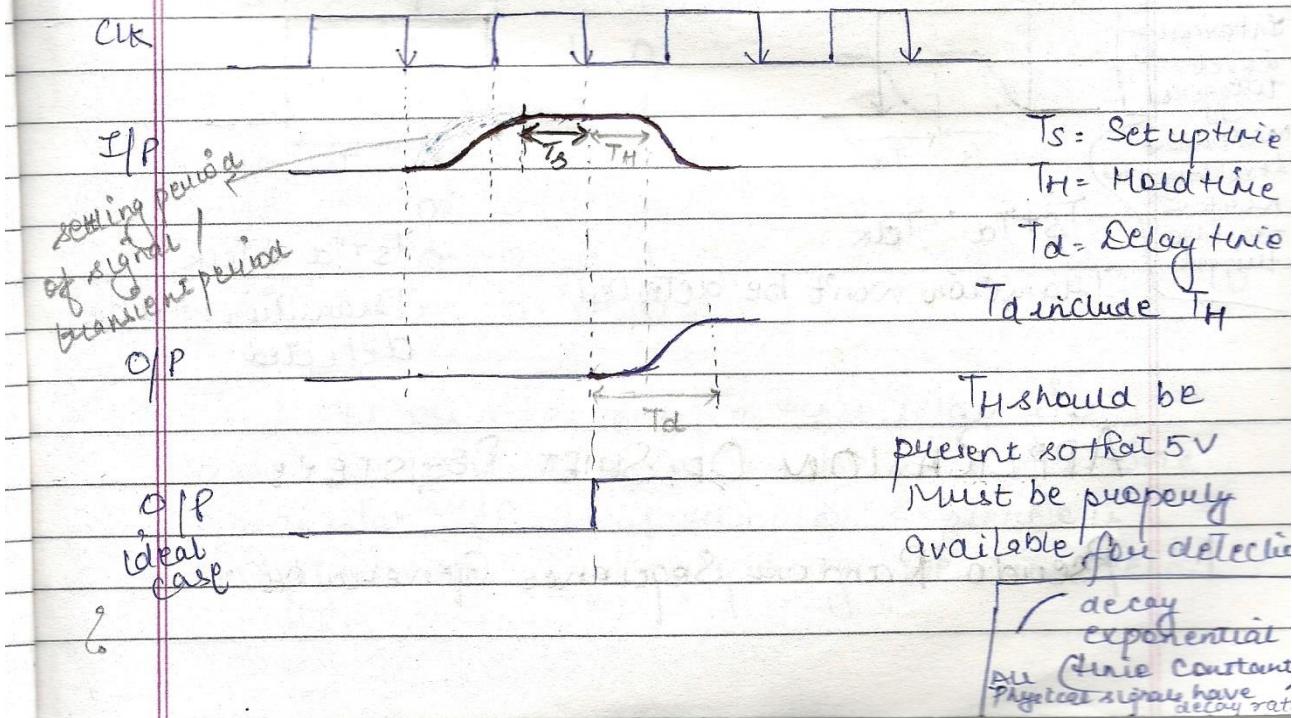
Shift register may be :-

- Shift right shift register
- Shift left shift register

Hence there are total 8 modes of operation

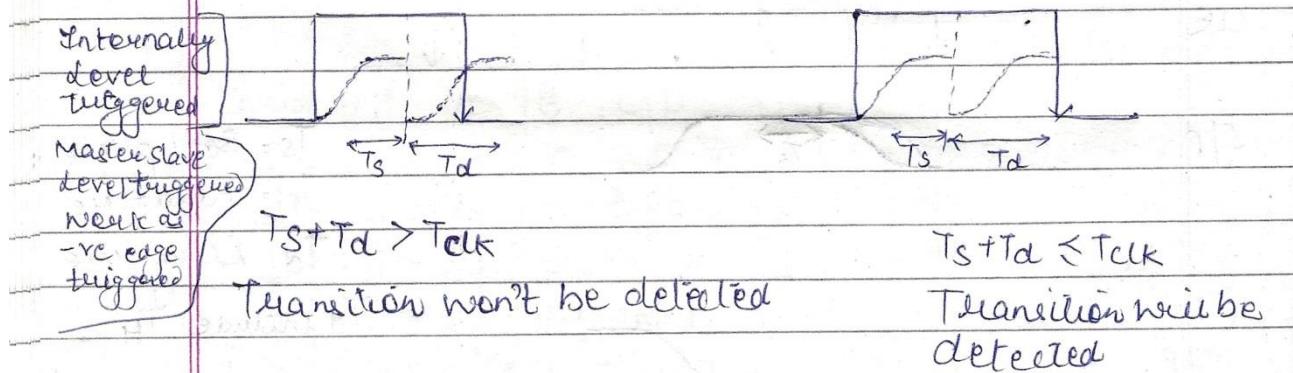
Universal Shift Register: Register that ^{can operate in} allows all the 8 modes.

CONCEPT OF SET UP TIME AND HOLD TIME:



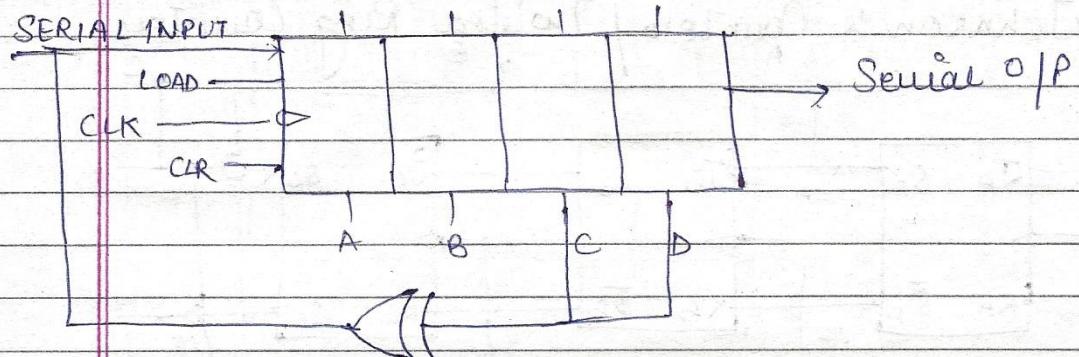
- Set up time - It is the minimum time required for the input to attain the stable value of 5V before the arrival of -ve edge of clock.
- After the -ve edge of clock the O/P should not be 0 at once but rather it should remain at 1 for specific period of time for reliable output.
- The minimum time for which the logic 1 is maintained after the -ve edge of clock is called hold time.
- The time taken by the O/P to attain logic 0 after the -ve edge of clock is called delay time.

$$T_{CLK} \geq T_s + T_d$$



APPLICATION OF SHIFT REGISTERS

1. Pseudo Random Sequence Generator :



Clock Period A B C D

0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	1	0	0	1
4	1	1	0	0
5	0	1	1	0
6	1	0	1	1
7	0	1	0	1
8	1	0	1	0
9	1	1	0	1
10	1	1	1	0
11	1	1	1	1
12	0	1	1	1
13	0	0	1	1
14	0	0	0	1
15	1	0	0	0

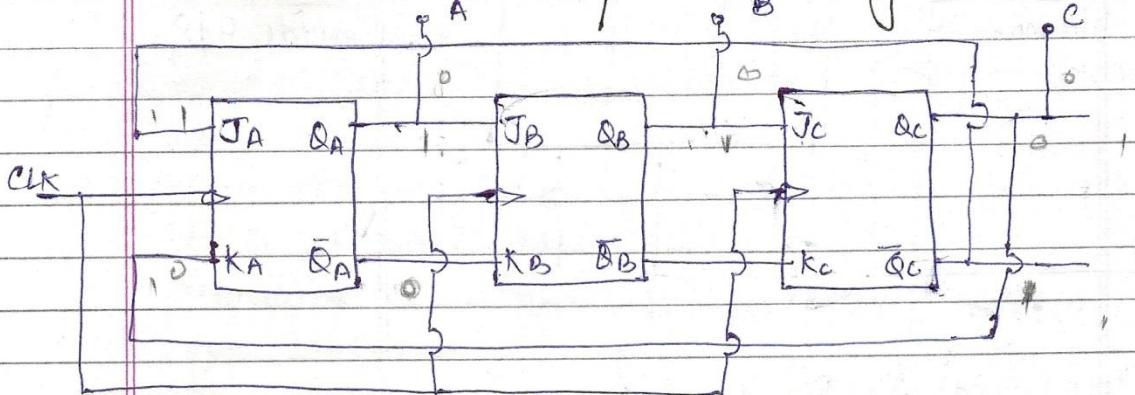
$$T = 15 T_{clk}$$

$$f = \frac{f_{clk}}{15}$$

It is named 80%:
data is random but
depends upon initial
data [well defined
O/P for particular
TIP]

Use: It provides longer time delay. It is used for error detection because for one particular TIP one particular sequence is defined. Thus by comparing we can check the errors.

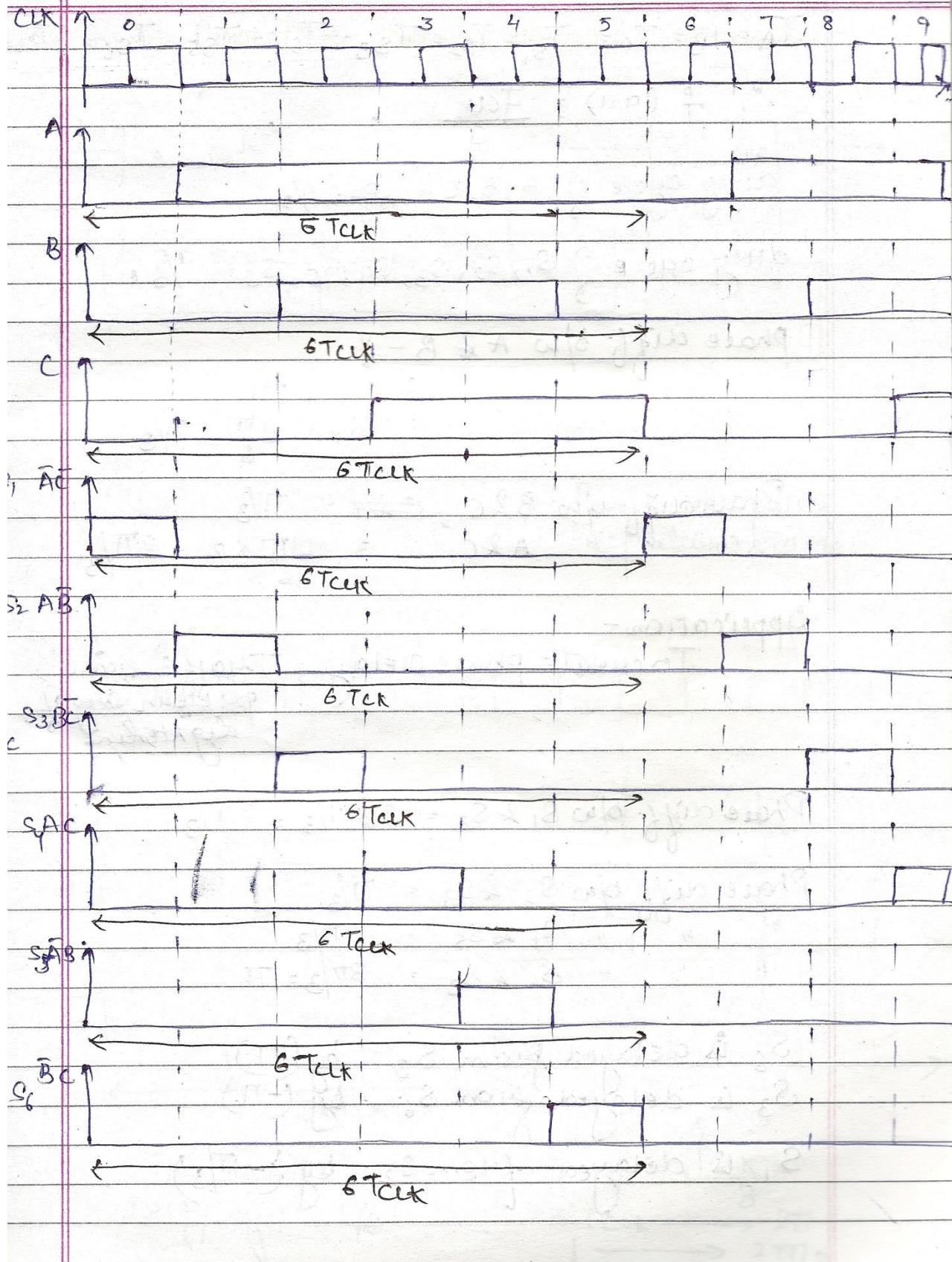
2. Johnson's Counter / Twisted Ring Counter



It is called twisted : inverted O/P is given to J & K of flip flop A.

Q Draw the timing diagram of $S_1 = \bar{A}\bar{C}$, $S_2 = A\bar{B}$, $S_3 = B\bar{C}$, $S_4 = AC$, $S_5 = \bar{A}B$, $S_6 = \bar{B}C$. Find the freq. of all signals and phase difference b/w S_6 & S_3 , S_1 & S_3 , S_2 & S_3 , S_1 & S_5

C.P.	A	B	C	\bar{A}	\bar{B}	\bar{C}	$\bar{A}\bar{C}$	$A\bar{B}$	$B\bar{C}$	AC	\bar{AB}
t=0	0	0	0	1	1	1	1	0	0	0	0
1	1	0	0	0	1	1	0	1	0	0	0
2	1	1	0	0	0	1	0	0	0	1	0
3	1	1	1	0	0	0	0	0	0	1	0
4	0	1	1	1	0	0	0	0	0	0	1
5	0	0	1	1	1	0	0	0	0	0	0
6	0	0	0	1	1	1	1	0	0	0	0
7	1	0	0	0	1	1	0	1	0	0	0
8	1	1	0	0	0	1	0	0	1	0	0
9	1	1	1	0	0	0	0	0	0	1	0



$$T_A = T_B = T_C = T_{S_1} = T_{S_2} = T_{S_3} = T_{S_4} = T_{S_5} = T_{S_6} = 6T_{CLK}$$

$$\therefore f(\text{clk}) = \frac{f_{clk}}{6}$$

$$\text{duty cycle of } A, B, C = \frac{3}{6} = \frac{1}{2}$$

$$\text{duty cycle of } S_1, S_2, S_3, S_4, S_5, S_6 = \frac{1}{6}$$

Phase diff. b/w A & B - ?

$$6T_{CLK} = 2\pi$$

$$1T_{CLK} = \frac{2\pi}{6} = \frac{\pi}{3}$$

$$\text{Phase diff. b/w B \& C} = \cancel{\pi/3}$$

$$\text{" " " " A \& C} = \frac{2\pi \times 2}{6} = \frac{2\pi}{3}$$

Application -

To create phase delay, Traffic signal, pattern of lighting

$$\text{Phase diff. b/w } S_1 \text{ \& } S_3 = 2 \times \frac{\pi}{3} = \frac{2\pi}{3}$$

$$\text{Phase diff. b/w } S_2 \text{ \& } S_3 = \frac{\pi}{3}$$

$$\text{" " " " } S_1 \text{ \& } S_5 = \frac{4\pi}{3}$$

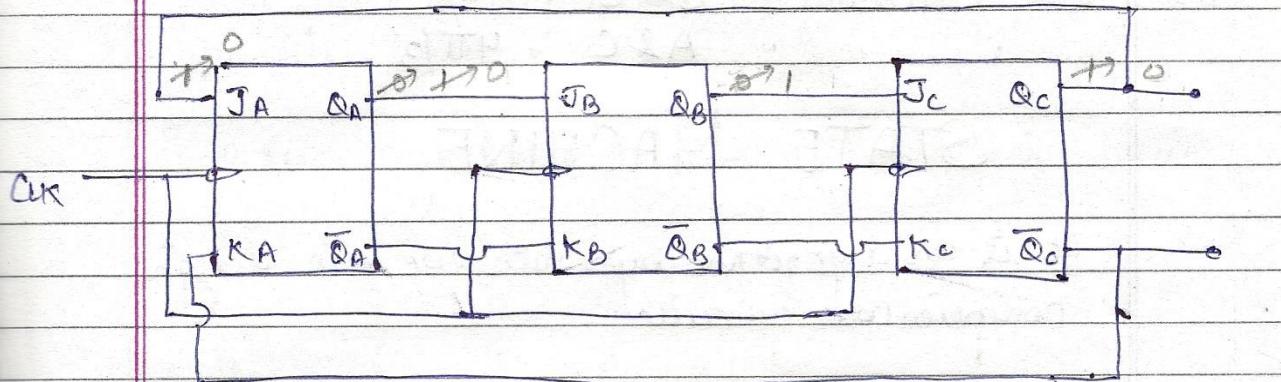
$$\text{" " " " } S_6 \text{ \& } S_3 = \frac{3\pi}{3} = \pi$$

S_6 is delayed from S_3 by (π)

S_3 is delayed from S_6 by $(-\pi)$

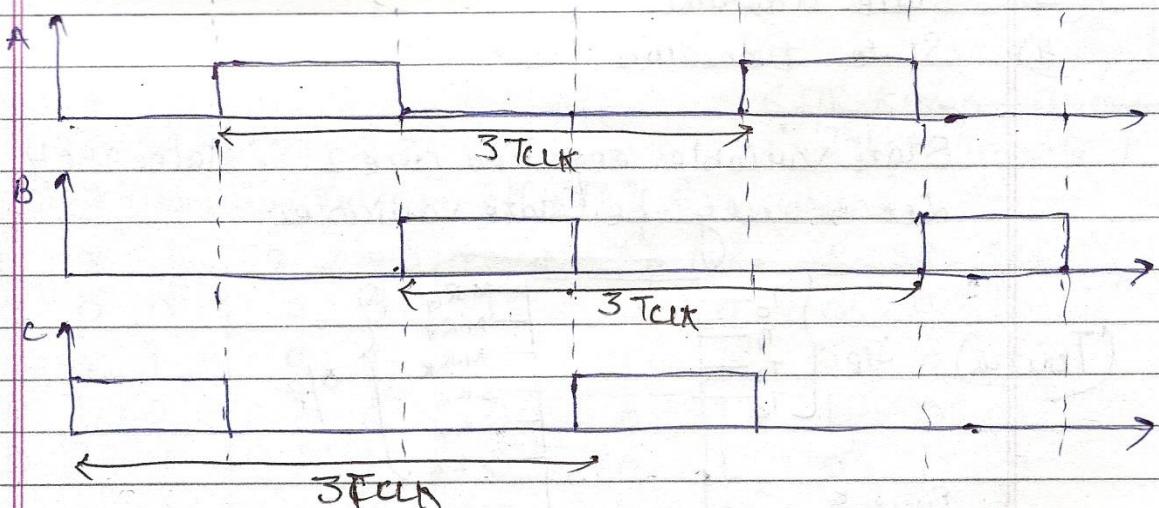
S_1 is delayed from S_2 by $(-\pi/3)$

3. RING COUNTER



C.P.	A	B	C
0	0	0	1
1	1	0	0
2	0	1	0
3	0	0	1

After 3 clock pulse
repetition starts



$$T_A = T_B = T_C = 3T_{CLK}, \quad f = \frac{f_{CLK}}{3}$$

Duty cycle 1/3

$$3T_{CLK} \xrightarrow{\text{---}} 2\pi \quad \xleftarrow{\text{---}} 2\pi/3$$

Date 19/6/18
Page

$$\therefore \text{Phase diff. b/w A \& B} = 2\pi/3$$

$$\therefore \quad \quad \quad \text{B \& C} = 2\pi/3$$

$$\therefore \quad \quad \quad \text{A \& C} = 4\pi/3$$

STATE MACHINE

It is an algorithm by which we can design complicated circuits.

e.g. Traffic light controller

	Main Road	Side Road
M.R.	MR	SR
S.R.	G R	
	Y R	
	R G	
	R Y	

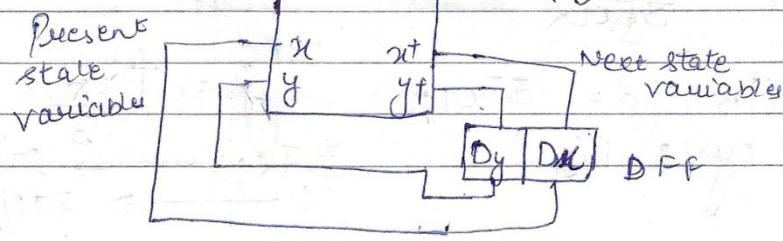
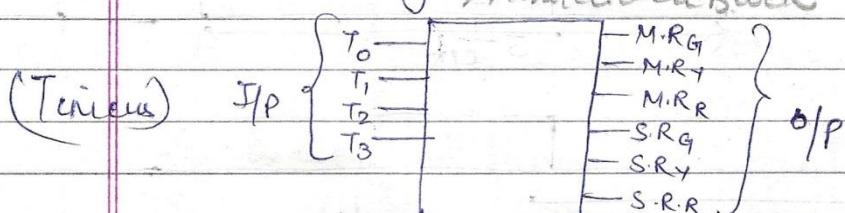
Steps followed are:-

- (1) Assign state variables
- (2) Design generalized architecture block
- (3) State diagram
- (4) State transition table

State variables required are 2 " states are 4

Let x and y be state variables.

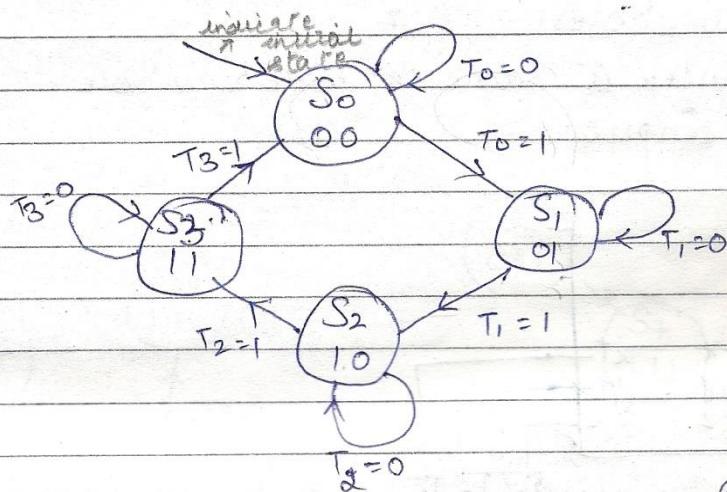
Architectural block



- T_1 O/P is 1 when T_0 is lapsed (complete)
- Next state is dependent on present variable and previous state

Q. - Flip flops are used to store the previous value.

I/P	M.R.	S.R.	x	y
T_0	G	R	0 0	S_0
T_1	Y	R	0 1	S_1
T_2	R	G	1 0	S_2
T_3	R	Y	1 1	S_3



$$I/P_s = 6 \left[\begin{array}{c} T_0 - T_4 \\ x, y \end{array} \right]$$

$$O/P_s = 8 \left[\begin{array}{c} M.R. \\ S.R. \\ x + y \end{array} \right]$$

6 I/P k-map - 64 cells
 \therefore Take $T_0 = T_1 = T_2 = T_3 = T$

State Transition Table - I/P, O/P & state relation

T	x	y	x^+	y^+	M.R. G	M.R. Y	M.R. R	S.R. G	S.R. Y	S.R. R
0	0	0	0	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0	0	0	1
0	1	0	1	0	0	0	1	1	0	0
0	1	1	1	1	0	0	1	0	1	0
1	0	0	0	1	0	1	0	0	0	1
1	0	1	1	0	0	0	1	1	0	0
1	1	0	1	1	0	0	1	0	1	0
1	1	1	0	0	1	0	0	0	0	1

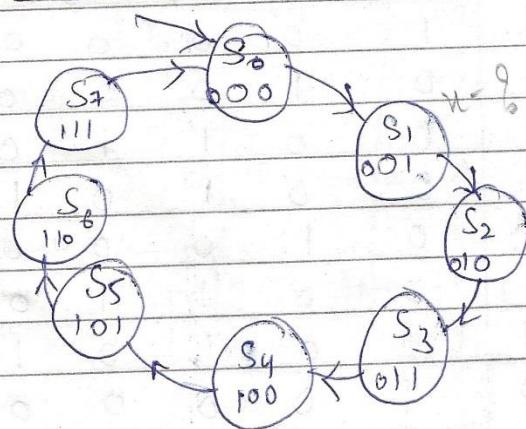
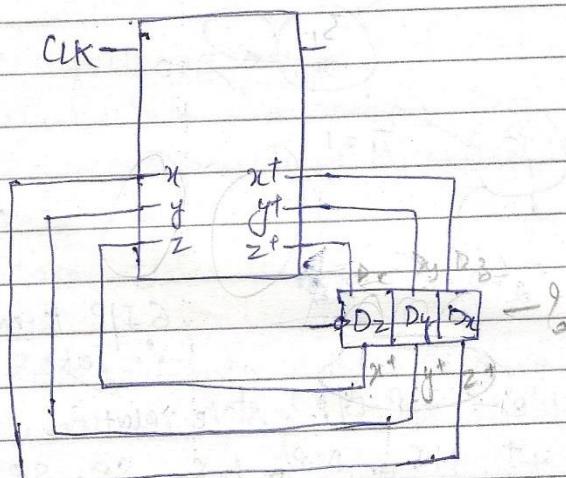
6 K Maps of ($2^3=8$) cells will be drawn to get minimised logic for the O/Ps

Pb Design 3 bit synchronous counter of count 0-7

- * State M/C is used for synchronous devices only.
- Asyn' devices are not used practically. The combination of two different sequential cells will cause unreliability at O/P

Total states 0-7 = 8

- (1) 3 state variables x, y, z
- (2) O/P of counter is same as state variable.
- (3) CLK is input.

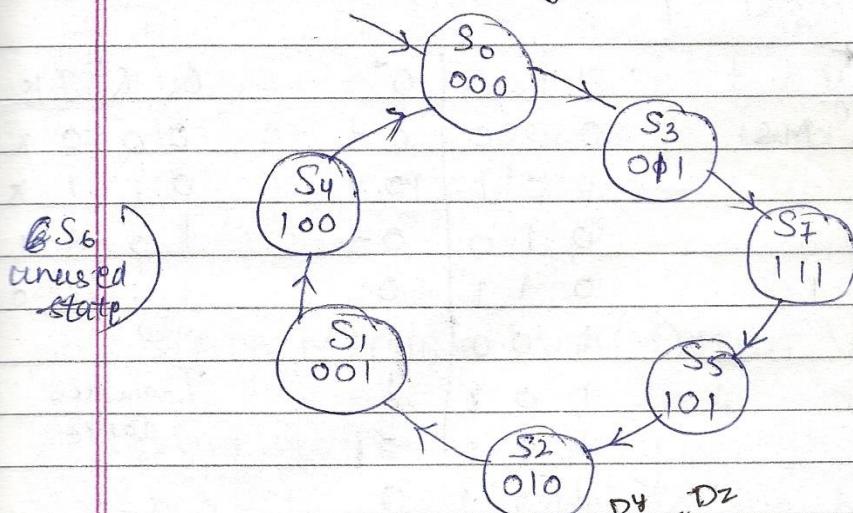


$ CLK \backslash x \cdot y \cdot z$	x^+	y^+	z^+
0 0 0 0	0	0	1
1 0 0 1	0	1	0
2 0 1 0	0	1	1
3 0 1 1	1	0	0
4 1 0 0	1	0	1
5 1 0 1	1	1	0
6 1 1 0	1	1	1
7 1 1 1	0	0	0

k-map (3) 8 cells
for x, y, z

Pb Design a counter $0 \rightarrow 3 \rightarrow 7 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 4$

1 State variables - 3 x, y, z



$ CLK \backslash x \cdot y \cdot z$	$x^+ \overset{Dx}{\sim}$	$y^+ \overset{Dy}{\sim}$	$z^+ \overset{Dz}{\sim}$	$J_x K_x$	$J_y K_y$	$J_z K_z$
0 0 0 0	0	0	1	0 x	1 x	1 x
1 0 0 1	0	1	0	1 x	0 x	x 1
2 0 1 0	0	0	1	0 x	x 1	1 x
3 0 1 1	1	1	1	1 x	x 0	x 0
4 1 0 0	0	0	0	x 1	0 x	0 x
5 1 0 1	0	1	0	x 1	1 x	x 1
6 1 1 0	x	x x	x	x x	x x	x x
7 1 1 1	1	0	1	x 0	x 1	x 0

- In case of D flip flops the same logic of x^+, y^+, z^+ can be used because $D_x = x^+$, $D_y = y^+$, $D_z = z^+$
 Q. but in case of J K flip flop we have to implement the logic of J & K and not of x^+, y^+, z^+ .

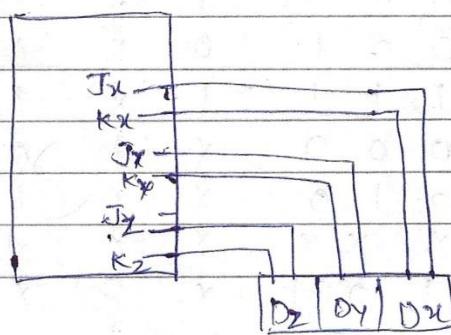
D	Q^+	D	D	Q	Q^+
0	0	0	0	0	0
0	1	0	1	0	0
1	0	0	1	0	1
1	1	1	1	1	1

i.e. D is always equal to Q

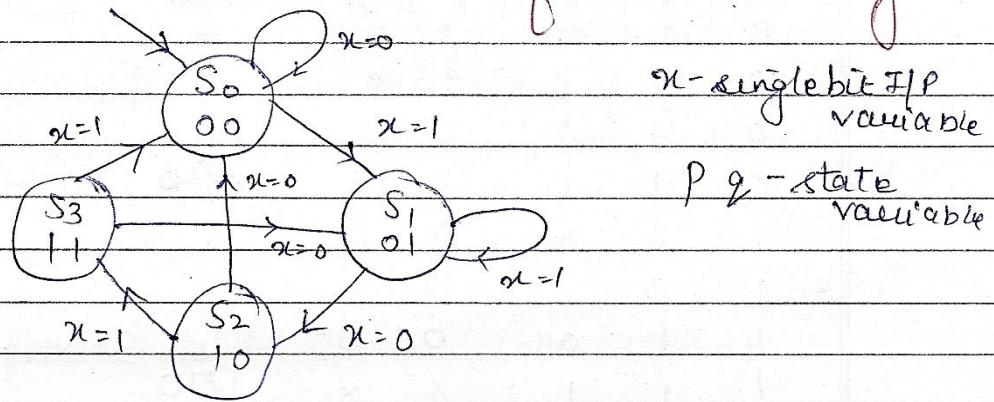
J	K	Q^+	J	K	Q	Q^+	Q	Q^+	J	K
0	0	Q M.S.	0	0	0	0	0	0	0	0 x
0	1	0	0	0	1	1	0	1	1	x
1	0	1	0	1	0	0	1	0	1	x
1	1	\bar{Q} T.	0	1	1	0	1	1	x	0
			1	0	0	1				
			1	0	1	1				
			1	1	0	0				
			1	1	1	0				

Transition
table

Expansion of Ch. Table



Pb Draw state transition table from state diagram



x	p	q	p^+	q^+	.
0	0	0	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	1	0	1	
1	0	0	0	1	
1	0	1	0	1	
1	1	0	1	1	
1	1	1	0	0	

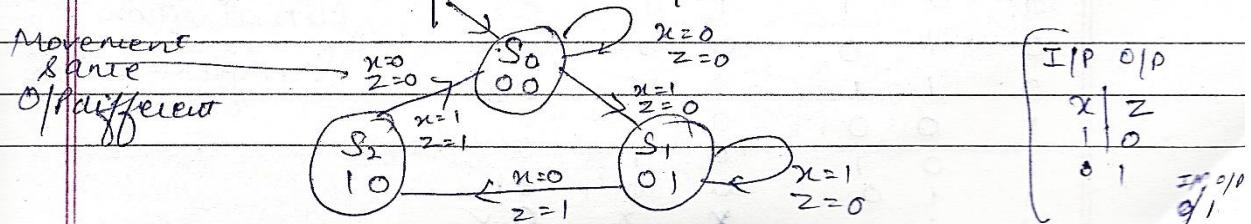
If we use D flip flops
then 2 K-maps for p^+ & q^+
but if JK flip flops are
used then 4 K-maps for
 Jp^+, Kp^+ & Jq^+, Kq^+

STATE MACHINE WITH OUTPUT VARIABLE

Based on the No. of output variables associated with the state machine we have two types of machines:-

- (1) Mealy Machine
- (2) Moore M/C

1 Mealy Machine - Outputs are defined for present state and depend on I/P condition.

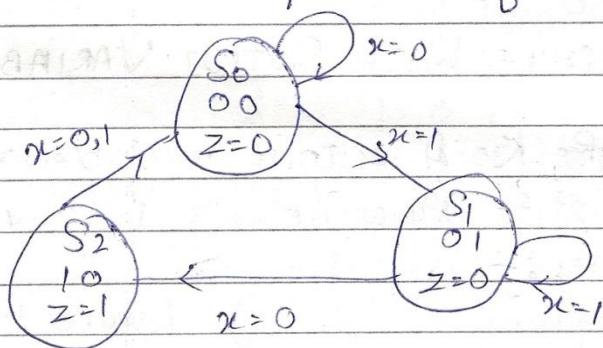


x	p	q	p^+	q^+	z
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	x	x	<u>$\boxed{0}$</u>
1	0	0	0	1	0
1	0	1	0	1	0
1	1	0	0	0	1
1	1	1	x	x	<u>$\boxed{0}$</u>

(cost minimization)

→ If don't care variable is used in O/P variable, then it will create problem so that why O/P variable will always be taken as 0. But state variable can be taken as don't care.

2: Mealy Machine - Outputs are defined for present state independent of I/F variables.



x	p	q	p^+	q^+	z
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	1	x	x	<u>$\boxed{0}$</u>
1	0	0	0	1	0
1	0	1	0	1	0
1	1	0	0	0	<u>$\boxed{1}$</u>
1	1	1	x	x	<u>$\boxed{0}$</u>

Minimization & Implementation

Application - Pattern detection : As soon as the pattern is detected the signal should be generated corresponding to the last bit of pattern.

Pb

Transmission line

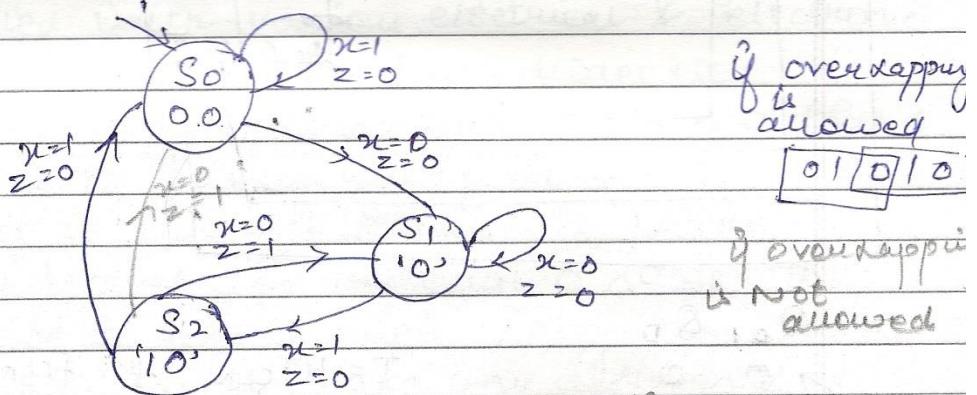
110101011'010110

Pattern \rightarrow 010 $z=1$

$S_0 \rightarrow$ Reset state

Main problem is to draw state machine

Mealy M/C

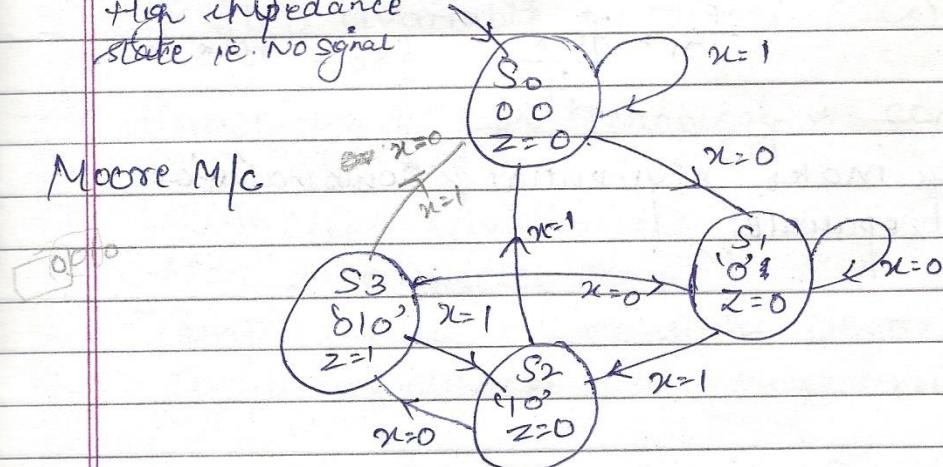


q.

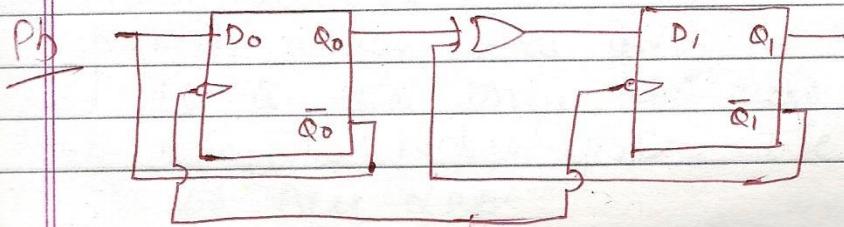
Reset state does not mean that S_0 is 0, it just simply means that we have started from S_0 .

High impedance
state i.e. NO signal

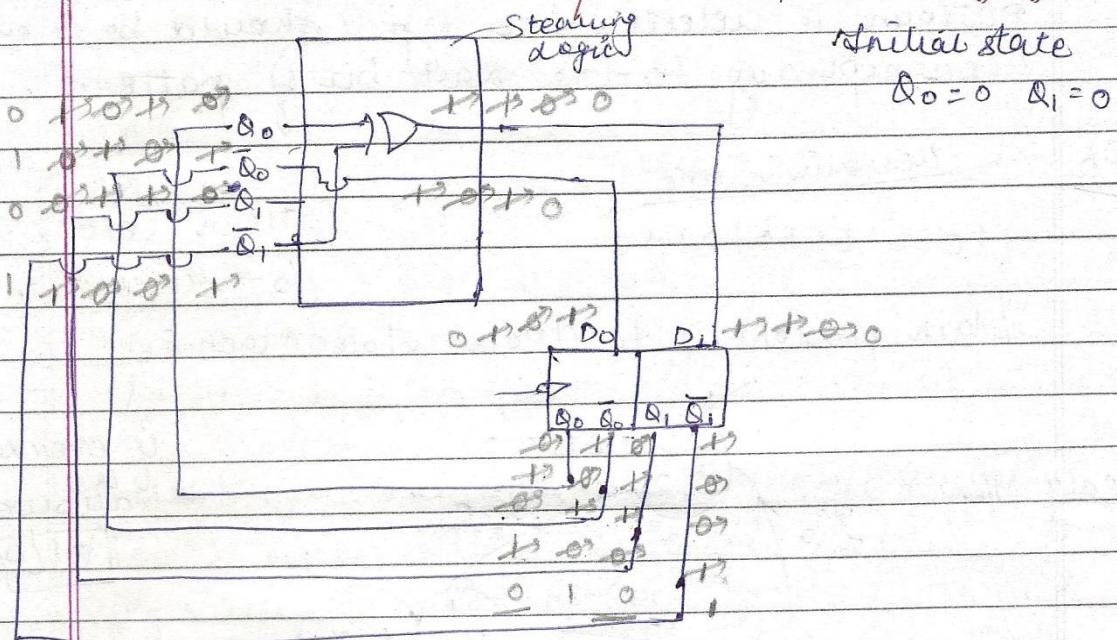
Moore M/C



Pb



Find State Transition Sequence, freq., duty cycle



Q_0, Q_1

0	0
1	1
1	0
0	1

$$T = 4T_{CLK} \quad f = f_{CLK}/4$$

$$\text{Duty cycle } Q_1 = 2/4 = 1/2$$

$$\text{, " } Q_0 = 2/4 = 1/2$$

$$f_{Q_1} = \frac{f_{CLK}}{4}, \quad f_{Q_0} = \frac{f_{CLK}}{2}$$

— Always make sequential & combinational parts separate.

MEDIUM SCALE INTEGRATION



Based on hardware:

SSI	< 10	gates equivalent ckt. in single package
MSI	10-100	"
LSI	100-1000	"
VLSI	> 1000	"

It is just based on software. There are just a single package on which no. of gates are integrated with various electrical & electronic components like transistors, diodes etc.

If the cost of single gate is 1 Re and that of 12 gates is 12 but if we purchase an IC of 50 gates it will be equal to 12.5 Re - 13 Re. This is because the material is same. The difference is only in their integration process and that hardly make any difference in their cost.

Even single gate is not available in market
↳ It is available in bunches (ICs)

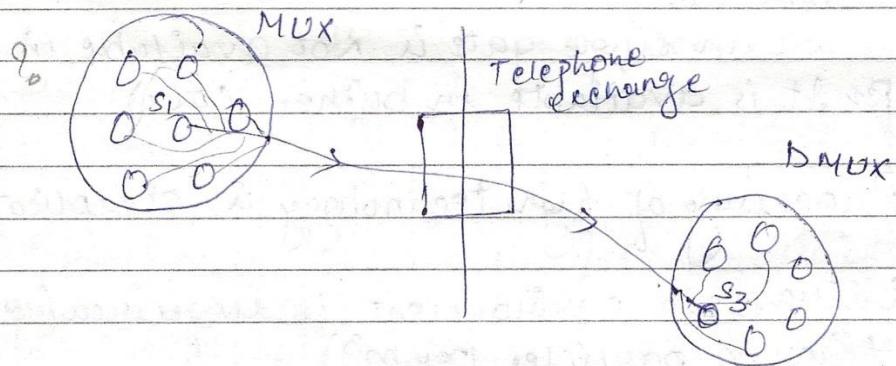
→ Processing of high technology is complicated.

- Ultra clean environment is required (less than 20 particles per cm^3)
- Temp. should be constant becoz the semi-conductor devices are very sensitive to temp.

→ Integration is done with the help of planar process which was given by John Kelvin. He is the only ~~scientist~~^{engineer} who had received Nobel prize. He worked in Bell Lab.

- The level of integration is high b. in high technology but processing, etching and oxidation is almost same
- Oxidation and etching is done for proper shaping of components on semiconductor chip after integration?
- There is no industry of semiconductor devices in India
- ~~AS~~ Integrated circuits are used because they are robust. More wiring more exposed to damage but in IC there is no such thing. Everything is internal.

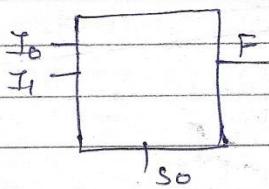
MSI - Multiplexer, demultiplexer, encoder, decoder. (20-50 gates)



Suppose S_1 has dialled the number of S_3 then telephone exchange will select the line connect the line S_3 and after that it will deliver the O/P at the line S_3 .

MULTIPLEXER - It is the selection of a line from no. of lines at a particular instant.

2:1 MUX

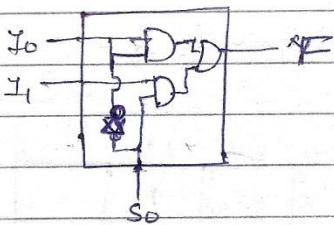


We have to select either of these two lines. For that 1 conditionals have to be there.

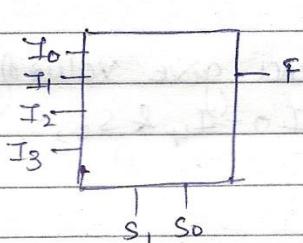
Two conditions are need to select two IfPs so only 1 select line is need.

S_0	F
0	I_0
1	I_1

$$F = \bar{S}_0 I_0 + S_0 I_1 \\ = 0 \cdot I_0 + 1 \cdot I_0 \\ = I_0$$



4:1 MUX



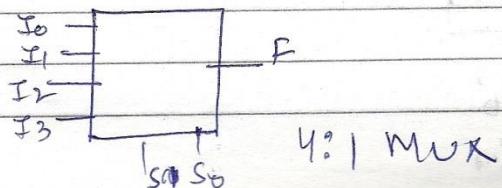
S_1	S_0	F
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$F = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

Power consumption is low in IC.

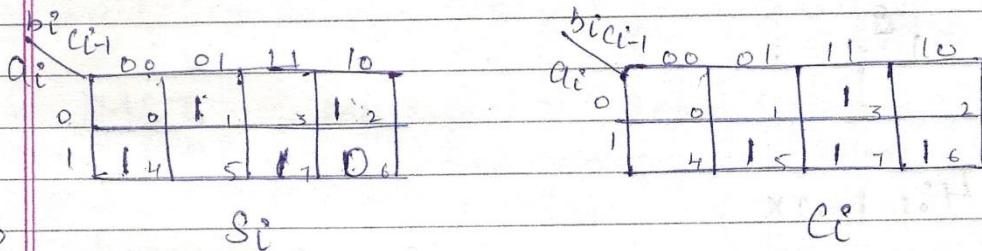
Application - Implement full adder by using 4:1 MUX

→ Two MUX in a single IC is called dual MUX



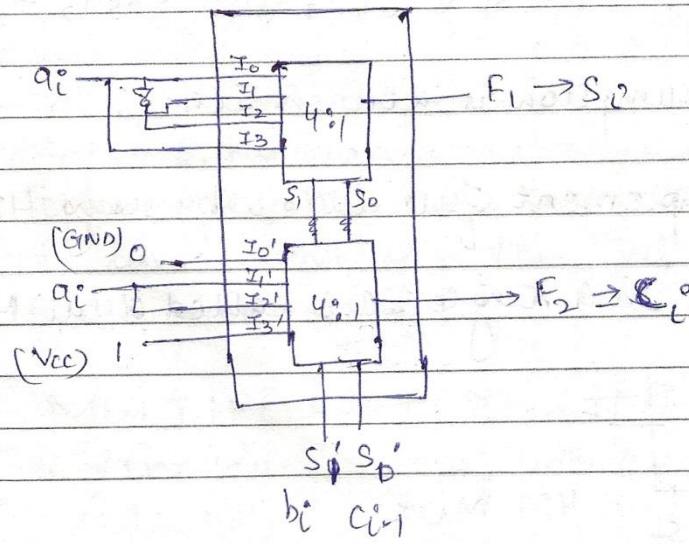
a_i^o	b_i^o	c_{i-1}^o	S_i^o	C_i^o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Two 4:1 MUX need to be used 1 for S_i^o & other for C_i^o



* don't do pairing

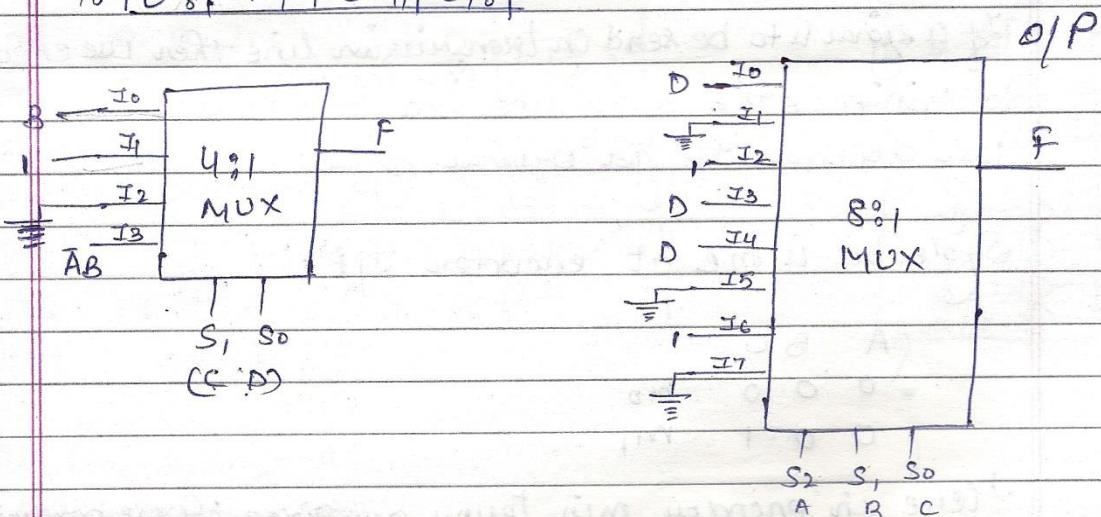
Take $b_i c_{i-1}$ on select lines and give value of a_i^o as I/P \therefore we have 6 I/P ($I_0 - I_4$ & S_0, S_1) but available a_i^o, b_i, c_{i-1} .



Pb Implement $F(A, B, C, D) = \sum m(1, 4, 5, 7, 9, 12, 13)$

a) 4:1 MUX b) 8:1 MUX

CD AB	00	01	11	10
00	0 ₀	1 ₁	0 ₃	0 ₂
01	1 ₄	1 ₅	1 ₇	0 ₆
11	1 ₁₂	1 ₁₃	0 ₁₅	0 ₁₄
10	0 ₈	1 ₉	0 ₁₁	0 ₁₀

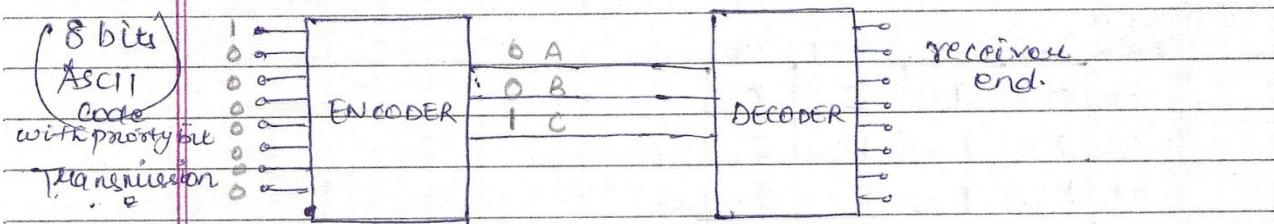


We can also take BCD on select lines

But by takes A BC cell will be adjacent.

- Question are asked in a way that O/P is given then minimize logic is asked.
- Draw k-map, do pairing possible, minimize logic. That will be O/P behavior.
- Demux is not used for implementation. It is used for practical application.

ENCODER / DECODER



If a signal is to be send on transmission line then use encoder to minimize the no. of lines and at receiver end use decoder to get original signal.

Single bit is one at encoder I/P:

A	B	C	
0	0	0	M ₀
0	0	1	M ₁

Here in encoder min terms are given. Corresponding to min term (which is 1) code at ABC is generated i.e. reverse process.

- If more than 1 input bit is '1' then priority encoder is used. But in this case exact O/P will not be generated by decoder.
- In priority encoder it generates code at A, B, C corresponding to priority min. term. Hence the other min term (which is 1)'s code won't be generated and hence can't be decoded by decoder.

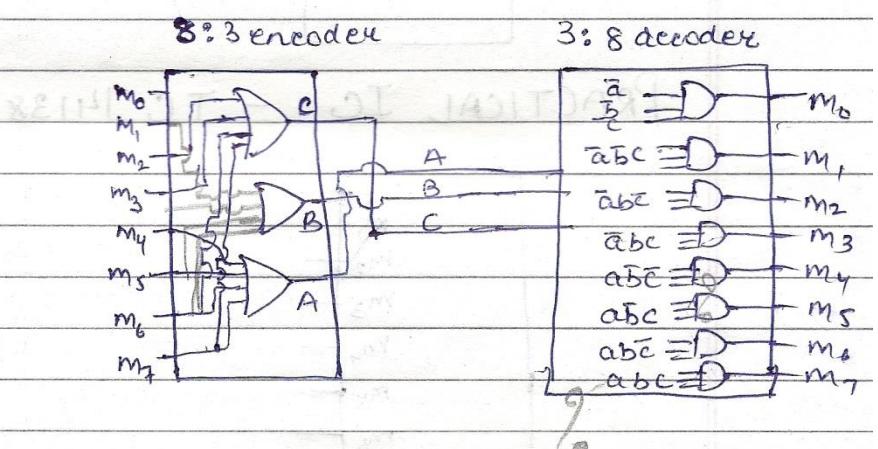
- 1st priority encoder generates code against min term which is 1, at first
- 2nd priority encoder generates code against min term.

which is 1 at last.

Since decision codes cannot be recovered. \therefore it is used for particular application only.

\rightarrow Encoder has single bit 1 at I/P

	A	B	C
m_0	0	0	0
m_1	0	0	1
m_2	0	1	0
m_3	0	1	1
m_4	1	0	0
m_5	1	0	1
m_6	1	1	0
m_7	1	1	1



$$m_4 + m_5 + m_6 + m_7 \rightarrow A$$

$$m_2 + m_3 + m_6 + m_7 \rightarrow B$$

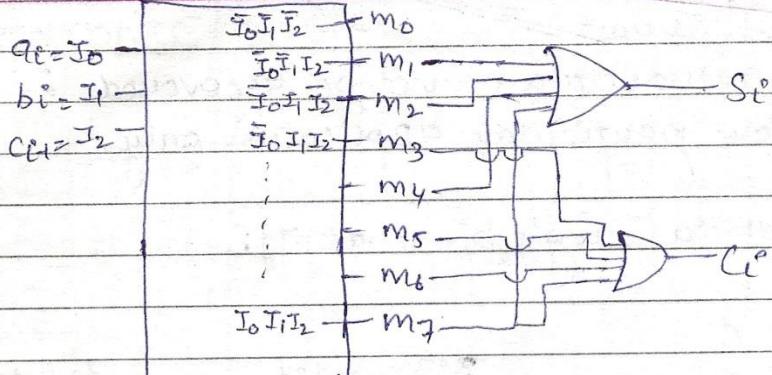
$$m_1 + m_3 + m_5 + m_7 \rightarrow C$$

Pb Implement full adder using 3:8 decoder

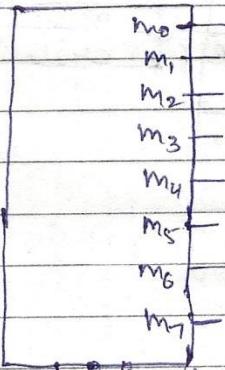
	a _i	b _i	c _{i-1}	S _i	C _i
m_0	0	0	0	0	0
m_1	0	0	1	1	0
m_2	0	1	0	1	0
m_3	0	1	1	0	1
m_4	1	0	0	1	0
m_5	1	0	1	0	1
m_6	1	1	0	0	1
m_7	1	1	1	1	1

$$S_i = \sum m(1, 2, 4, 7)$$

$$C_i = \sum m(3, 5, 6, 7)$$



PRACTICAL IC - IC 74138



$G_1 \rightarrow$ active high

$\bar{G}_2A \rightarrow$ active low

$G_2B \rightarrow$ active high

$G_1 \quad \bar{G}_2A \quad G_2B$

Here rather than 1 enable 3 are used so
as to activate IC corresponding to particular IP.

$G_1 \quad \bar{G}_2A \quad \bar{G}_2B$

0 0 0

0 0 1

0 1 0

0 1 1

If will be
enabled

1	0	0
---	---	---

1 0 1

1 1 0

1 1 1

but if we want IC to
be enabled at some

other value. Then

take PQR

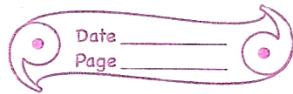
for eg - enable at 010

P Q R

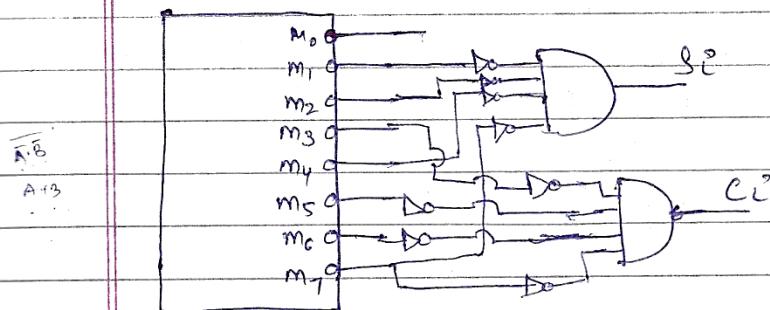
$\downarrow \quad \downarrow \quad |$
 $G_1 \quad \bar{G}_2A \quad \bar{G}_2B$

$0V \rightarrow 0$ $1 \rightarrow 5V$
 inverter

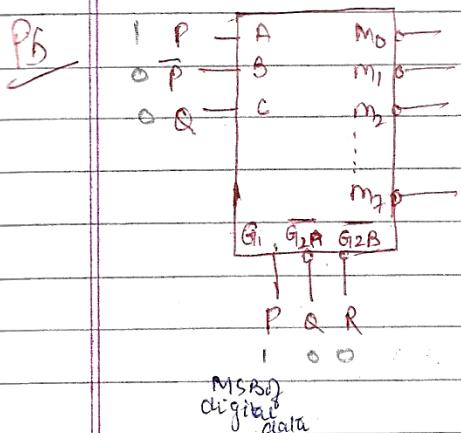
$0V \rightarrow 1$
 Active low



Inverted I/P at
 AND gate gives
 OR operation



- PQR is controlling ckt. Before writing four O/P check whether it is enabled or not.



$$F = \sum m(4, 5, 7)$$

V/q signal $v(t)$ is equal to

$$v(t) = 8t \quad 0 < t < 1 \text{ ns}$$

& digitised by using an ideal sampler having sampling interval of 0.1 ns and 3 bit quantiser is PQR. Find O/P of given ckt. at $t = 0.6 \text{ ns}$

$$\text{at } t = 0.6 \text{ ns} \text{ value } 8 \times 0.6 = 4.8$$

$$\text{Quantisation interval} = \frac{\text{Max. value}}{2^3} = \frac{8}{8} = 1$$

$$\therefore 4.8 \text{ will truncate to } 4 = L_4 = 100$$

100 is given to PQR

∴ IC is enabled

At ABC value is 100 (P̄PQ)

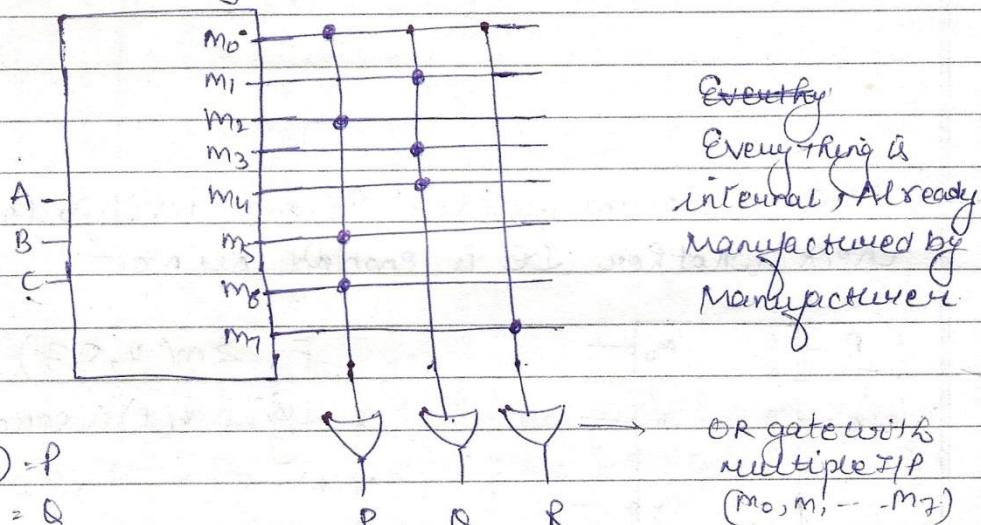
Since D/P is 1 at M₄

∴ Ano o/p is 1

LSI (LARGE SCALE INTEGRATION)

ROM: Read Only Memory.

It is fixed etched Hardware Truth Table
non-programmable



$$m(0, 2, 5, 6) = P$$

$$m(1, 3, 4) = Q$$

$$m(7) = R$$

OR gate with multiple I/P
(m₀, m₁, ..., m₇)

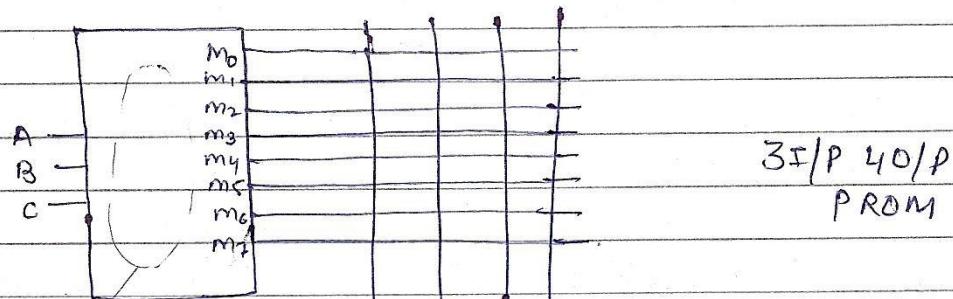
Suppose connections are given by manufacturer as shown in fig

We can use this particular IC for a particular O/P

In case we want some other O/P then it cannot be used. Hence it is not reliable.
External HW needs to be used for other O/P

Since we can't change any connections \Rightarrow it is called Read Only Memory and is non-programmable.

PROM :-



AND-ARRAY
(fixed)

W X Y Z

OR-ARRAY
(programmable) i.e. Connections to be made by user

This type of hardware is known as firm ware.

There is a provision by which I can treat/erase the connections made for particular application and reprogram the PROM. That type of PROM is Erasable PROM (EPROM).

EPROM $\xrightarrow{\text{UV}} \text{UVEPROM}$

$\xrightarrow{\text{EE}} \text{EEPROM}^{\checkmark}$ [electrically erasable PROM]

For internal connection of gates in IC the techniques used are :-

- Fusing technique and anti-fusing technique (Old)
- Charging " and discharging " (New)
[potential high / low]

→ Most recent is EEPROM & electrically alterable PROM or flash memory. (like LASER)

$m(1,7) \& m(1,2)$

same

In this connection which is same is kept as such others are changed.

Pb Implement 4 bit binary to Gray code converter by using PROM

Solu: v. Imp

Specifications :-

→ 4 I/Ps, 4 O/Ps PROM

4 I/P - 4 bit I/P generates 16 min terms
($m_0 - m_{15}$)

→ Size of PROM = $2^N \times M$

$N \rightarrow$ No. of I/P

$M \rightarrow$ No. of O/P

$$= 2^4 \times 4 = 64 \quad [\text{Total No. of connections}]$$

→ PROM from content table (in HEX code)

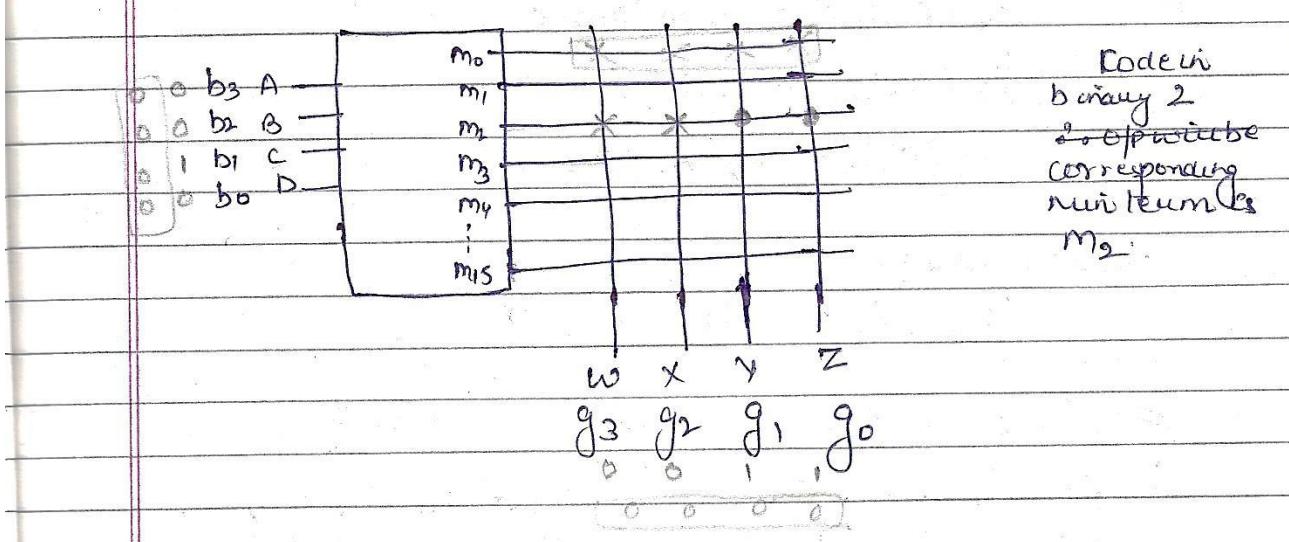
disadvantages of PROM

→ Size increases exponentially with no. of I/Ps (2^N)
and linearly with no. of O/Ps

Moreover AND array is not programmable
Hence ~~it is~~ Hardware is more

3 I/Ps to each AND gate. Total
AND gates = 16

Address in		Hexcode	g_3	g_2	g_1	g_0	Hex code
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1
0	0	2	0	0	1	1	3
0	0	3	0	0	1	0	2
0	1	4	0	1	0	0	6
0	1	5	0	1	1	1	7
0	1	6	0	1	0	1	5
0	1	7	0	1	0	0	4
1	0	8	1	1	0	0	C
1	0	9	1	1	0	1	D
1	0	A	1	1	0	1	F
1	0	B	1	1	0	0	E
1	1	C	1	0	0	1	A
1	1	D	1	0	1	1	B
1	1	E	1	0	0	1	9



Ques asked due: Connection is given and address

3 corresponding content is asked?

3 → 0011 corresponding number m₃

on m₃ line check connection and hence corresponding content.

PROM programmer - It is a device to program PROM. PROM programmer is interfaced with PC. Then PROM device (H/W) is inserted in PROM programmer and programming is done.

(Not in syllabus)

(2) PLA (Programmable Logic Array)

In this ~~array~~^{PROM} both AND-Array and OR-array is programmable.

Hence we use product terms in AND array rather than minterms to reduce hardware.

For this minimum logic need to be found using K-Map.

~~PB~~

$$f_1(a, b, c) = \sum m(0, 2)$$

$$f_2(a, b, c) = \sum m(0, 3, 4)$$

$$f_3(a, b, c) = \sum m(3, 4, 5, 7)$$

$$f_4(a, b, c) = \sum m(0, 3, 4, 7)$$

Specifications :-

$$1 \quad Y/P_x = 3, \quad 0/P_y = 4$$

2 To find product terms solve K-map

a\bc	00	01	11	10
0	1	0	1	1
1	1	1	0	0

a\bc	00	01	11	10
0	1	0	1	1
1	1	1	0	0

$$P_0 = \bar{a}\bar{c} = f_1$$

$$P_1 = \bar{b}\bar{c} + \overline{abc}$$

	bc	00	01	11	10
a	0	0	1	3	2
	1	14	18	17	6

$$f_3 = \overline{ab} + bc$$

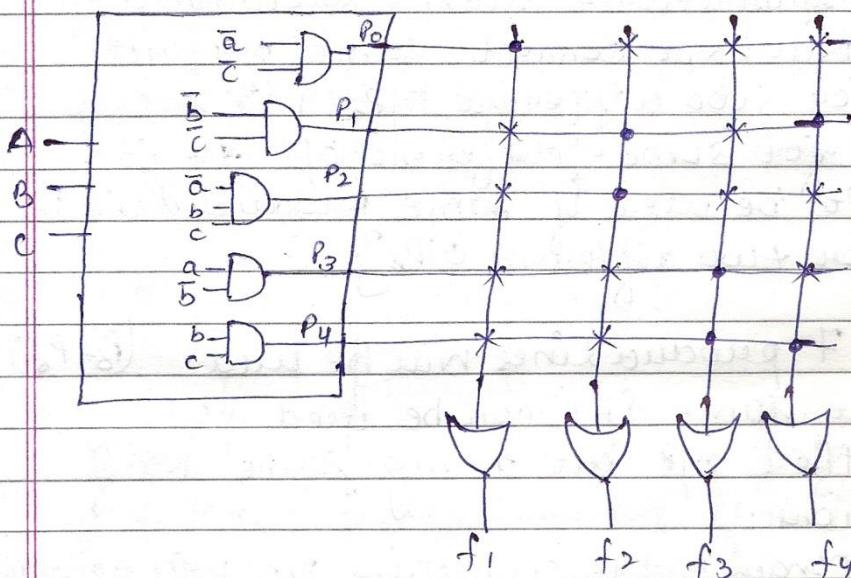
	bc	00	01	11	10
a	0	1	0	1	3
	1	14	5	17	6

$$f_4 = \overline{b}\overline{c} + bc \rightarrow P_4$$

* * * Repetition

Same product term can be shared. DR array is programmable.

∴ Product terms are 5
(P₀ - P₄)



Product lines can be shared by different O/Ps.

3 Size of PLA = P × M P → No. of product terms
M → No. of O/Ps

∴ Size of PLA is much less than PROM
⇒ Programming complexity increases but T/W decreases

(3) PAL - (Programmable Array logic)

It is similar to PLA in ckt wise.
AND Array is programmable but OR array is fixed.

OR side connections are already done by manufacturer.

Pb

same as previous

$$\text{Hence } f_4 = \bar{B}\bar{C} + BC$$

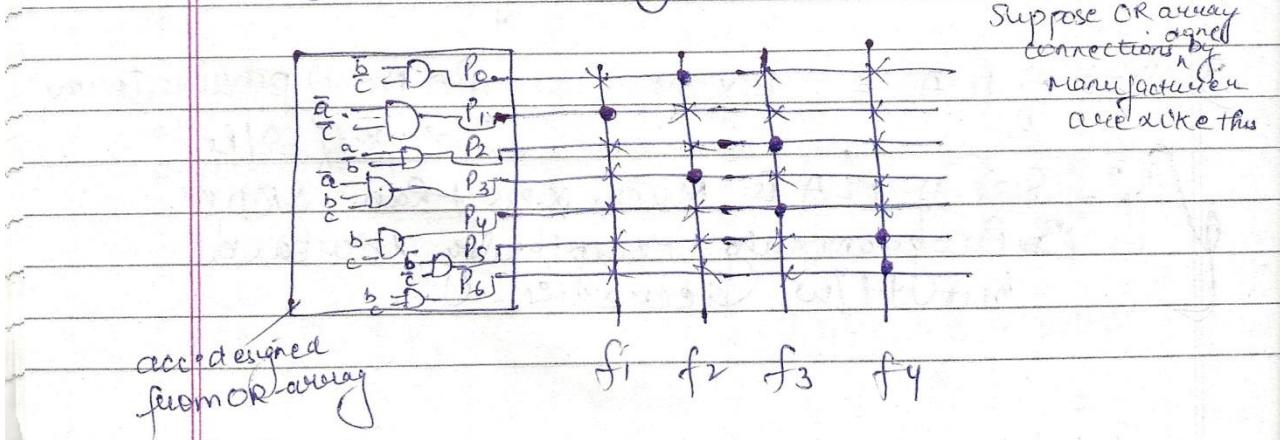
which is repetition

→ Since manufacturer doesn't know about O/P
∴ he will not connect same product
line for two different O/Ps [∴ in case
of use for some other different O/P so it
couldn't be used if same product line is
kept for two different O/Ps]

→ Hence 7 product lines will be used. ($P_0 - P_6$)

so that same PAL can be used for
some other O/P but having same no. of
connections.

→ Corresponding to the connections we will accordingly
design AND array



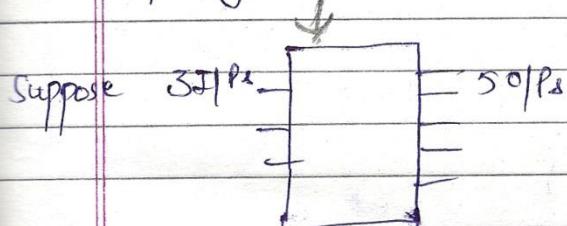
Specifications :-

- I/P i.e. N = 3, O/P i.e. M = 4, Product terms = 7
- Size = $P \times M$

- Size of PAL is more than PLA [if repetition exists]
- Programming complexity is less than PLA [less OR array
(is already manufactured)]
- Hardware complexity is less as compared to PROM.

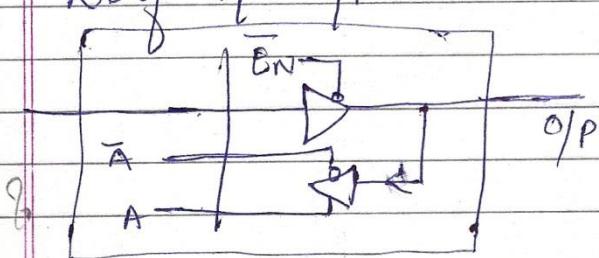
Specifications

- O/P is active low or active high
- OR O/P or X-OR O/P
- programmable I/O



but if we require 4 I/P and 4 O/P ckt. then
above ckt. cannot be used if it is unprogrammable
I/O ckt.

But if programmable then it can be used for any
No. of I/P O/P



Q

Design a a) Mealy M/c b) Moore M/c for a ct* with a serial I/P x and O/P z . If x is sampled in every clock transition, z should be 1 coinciding with the last bit of pattern 1001 or 010 otherwise O/P should be zero.

- (i) if overlapping is allowed
- (ii) if overlapping is not allowed

* Overlapping can occur only after completion of a pattern