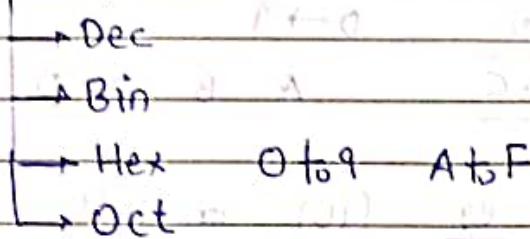


## Number-System



{Base → no. of distinct numbers in a number system}  $\downarrow$   
 levels

$$\dots n_3 n_2 n_1 n_0 \cdot n_{-1} n_{-2} n_{-3}$$

$$n_0 = (\text{Base})^0$$

$$n_1 = (\text{Base})^1$$

$$n_{-1} = (\text{Base})^{-1}$$

$$n_2 = (\text{Base})^2$$

$$n_{-2} = (\text{Base})^{-2}$$

### Binary

$$\text{Bin} \rightarrow 2 \swarrow \nwarrow$$

$$(0)_2, (1)_2, (10)_2, (11)_2, (100)_2$$

$$\dots n_3 n_2 n_1 n_0 n_{-1} n_{-2} n_{-3} \dots$$

$$n_0 = 2^0$$

$$n_{-1} = 2^{-1}$$

$$n_1 = 2^1$$

$$n_{-2} = 2^{-2}$$

$$n_2 = 2^2$$

$$n_{-3} = 2^{-3}$$

$$(10)_2 \rightarrow (?)_D \quad D = 2^0 \times 0 + 2^1 \times 1$$

$$\begin{matrix} 1 & 0 & \\ \downarrow & \downarrow & \\ x_0 & x_1 & \\ \downarrow & \downarrow & \\ 2^0 & 2^1 & 2^{-1} \end{matrix}$$

$$D = 2^1 \rightarrow (2)_D$$

Hexa Dec  
Dec + G

$0 \rightarrow 9$

A B C D E F

eg  $(10)_H$  or  $10_H$

$10_H = (\text{_____})_{(16)}$

$10_H = (16)_E$  or 16<sub>E</sub>

### Base - 5 number system

levels : 0 1 2 3 4

Next :  $(10)_5$

$$= 5^1 5^0 = 5$$

### Conversion of Number Systems

eg :  $(23)_{10} = (?)_2$

$$\begin{array}{r|l} 23 & 1 \\ 11 & 0 \\ 05 & 0 \\ 02 & 0 \\ \hline & 1 \end{array} \quad \begin{array}{r} (11001)_2 \\ (10111)_2 \end{array}$$

$$\begin{array}{r|l} 23 & 2 \\ 27 & \hline \end{array} \quad (27)_8$$

$$\begin{array}{r|l} 23 & 1 \\ 17 & \\ \hline 3 & \end{array} \quad (17)_{15}$$

$$(43)_5$$

decimal (whole part)

- Divide number with base of number system to be converted.
- Multiply fractional part with base.

$$(ABBA \cdot BAD)_{16} = (?)_{10}$$

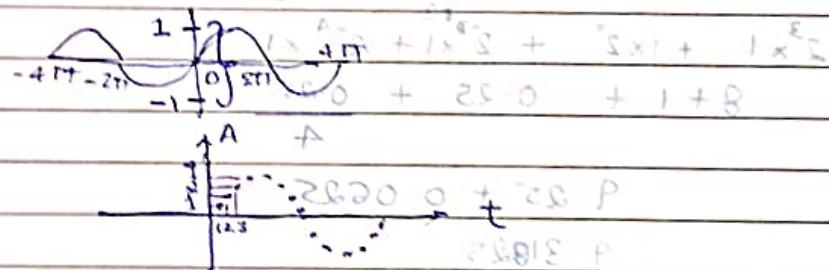
$$A = 10$$

$$B = 11$$

$$(10 \ 11 \ 10 \ 10 \cdot 11 \ 10 \ 13)_{16}$$

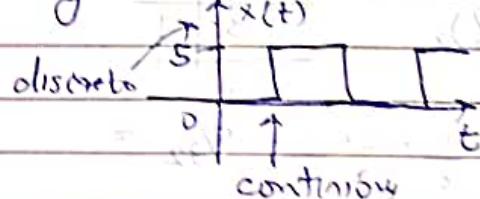
$$10 \times 16^3 + 11 \times 16^2 + 10 \times 16^1 + 10 \times 16^0 + 11 \times 16^{-1} + 10 \times 16^{-2} + 13 \times 16^{-3}$$

$$= (43962.729)_{10}$$



\* b/w If amplitude axis has finite value signal  
is digital

Digital  $\circ$  Continuous Time



### Boolean Algebra

$1+1 = 1$	→ Addition logic/ OR logic
$1+0 = 1$	
$0+1 = 1$	
$0+0 = 0$	

$1 \cdot 1 = 1$	→ Multiplication logic/ AND logic
$1 \cdot 0 = 0$	
$0 \cdot 1 = 0$	
$0 \cdot 0 = 0$	

1)  $(1001 \cdot 0101)_B \rightarrow (?)_D$

$$2^3 \times 1 + 1 \times 2^0 + 2^{-2} \times 1 + 2^{-4} \times 1 \\ 8 + 1 + 0.25 + \frac{0.25}{4}$$

$$9.25 + 0.0625$$

$$\underline{9.3125}$$

2)  $(13)_D = (?)_B$   
 $(1101)_B$

13	1
6	0
3	1
1	

3)  $(0.63625)_{10} = (?)_B$

$$\frac{6}{2} \times \frac{1}{2} + \frac{3}{4} \times \frac{1}{4} + \frac{6}{8} \times \frac{1}{8} + \frac{2}{16} \times \frac{1}{16} + \frac{5}{32} =$$

(X)

$$0.43625 \times 2 = 1.2725$$

$$0.2725 \times 2 = 0.545$$

$$0.545 \times 2 = 1.09$$

$$0.101$$

$$4) (6327 \cdot 4051)_8 = (?)_{16}$$

$$00 \cdot 01 = 01 \times 01$$

$$\begin{array}{r} 8 \\ | \quad 7 \times 8^3 + 2 \times 8^2 + 3 \times 8^1 + 0 \times 8^0 \\ + 0 \times 8^3 + 0 + 5 \times 8^2 + 1 \times 8^1 \\ \hline 3287 \cdot 51001 \end{array}$$

$$5) (3287 \cdot 510048)_{10} = (?)_2$$

$$\begin{array}{r} 3287 \mid 7 \\ \hline 410 \quad 2 \\ \hline 50 \\ \hline 51 \quad 3 \\ \hline 63 \\ \hline 0 \end{array} \quad \begin{array}{l} 0 \cdot 510048 \times 5 = 408032 \\ 0 \cdot 80384 \times 8 = 0 \cdot 64302 \\ 6327 + 0 \cdot 405 \\ \hline 6327 \cdot 4051 \end{array}$$

(?)  $\leftarrow$  (1111 0101 1001 101100)

$$6) (472)_8 = (?)_2$$

21 01 P S

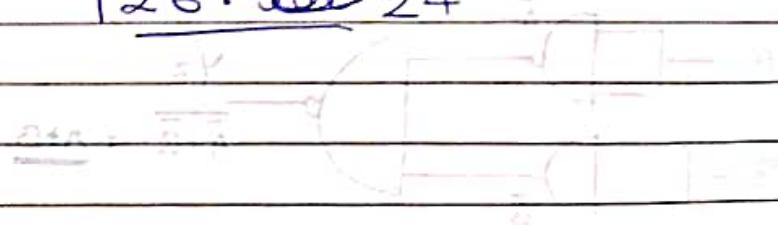
10011101010

$$(100111010)_{B^{10}} = ?_{2^{10}}$$

0.2

$$7) \underline{\underline{00(010110 \cdot 01010)_{B}}} = (?)_{10 \text{ oct}}$$

$126 \cdot 500 \cdot 24$



$$\textcircled{8} \quad (3A \cdot 2F)_{16} = (?)_D$$

$$16 \times 3 + 10 + 2 \times \frac{1}{16} + 15 \times \frac{1}{16^2}$$

$$= 58.1835$$

$$\textcircled{9} \quad (675 \cdot 625)_{10} = (?)_{16}$$

$$\begin{array}{r} 675 \\ \times 625 \\ \hline 3250 \\ 4200 \\ \hline 4125 \end{array}$$

$$0.625 \times 16 = 10.00$$

$$8(?) = 2A3 \cdot A$$

$$\textcircled{10} \quad (2F9A)_{16} = (?)_8$$

$$(10111001010)_B \rightarrow (57632)_8$$

$$\textcircled{11} \quad (001010011010111)_B \rightarrow (?)_H$$

2 9 10 15

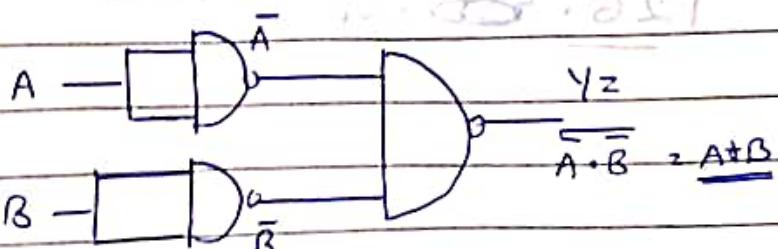
$$(29AF)$$

$$101011011001$$

$$101011001$$

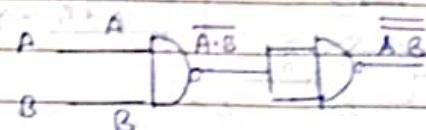
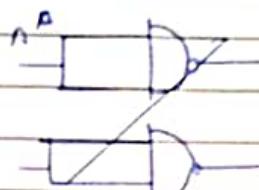
NAND

$$\text{as OR } Y = \overline{\overline{A} + \overline{B}} = \overline{\overline{A}} \cdot \overline{\overline{B}}$$



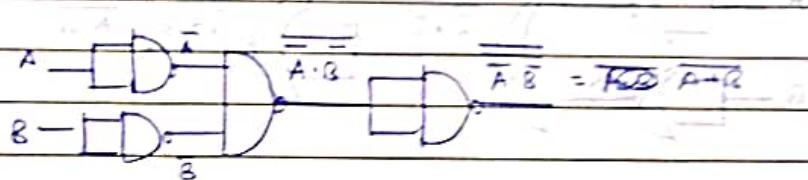
as AND

$$Y = \overline{A+B} = \overline{A} \cdot \overline{B}$$



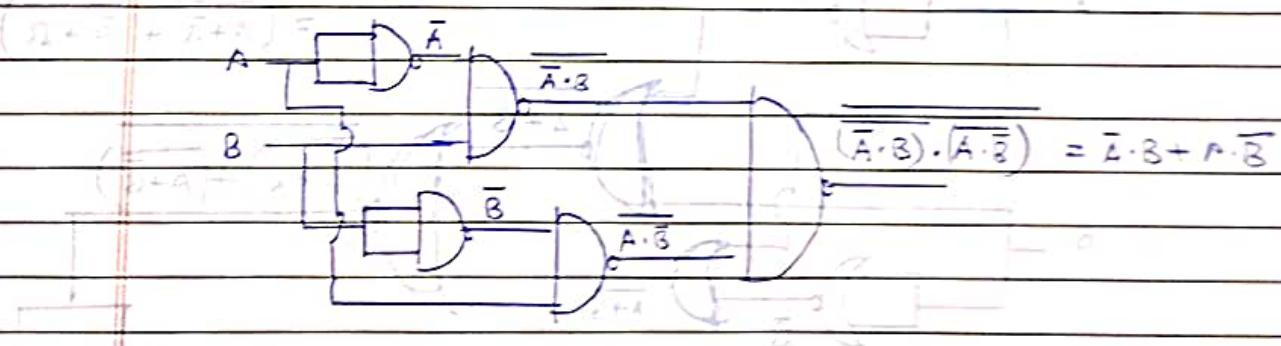
as NOR

$$Y = \overline{\overline{A+B}} = \overline{\overline{A} \cdot \overline{B}}$$



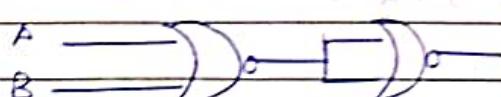
$(A+\overline{A}) \cdot (\overline{B}+\overline{B})$  as EXOR

$$(A+\overline{A}) \cdot (\overline{B}+\overline{B}) = Y = \overline{A \cdot B} + A \cdot \overline{B} = (\overline{A} \cdot \overline{B}) + (A \cdot \overline{B})$$

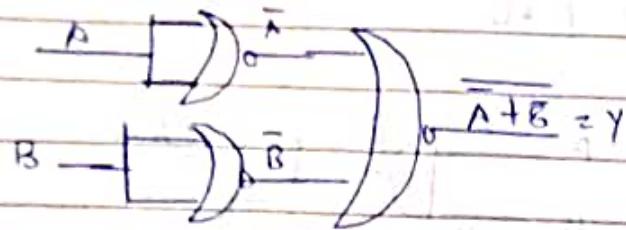


NOR

$$\text{as OR } Y = A+B = \overline{\overline{A} \cdot \overline{B}}$$

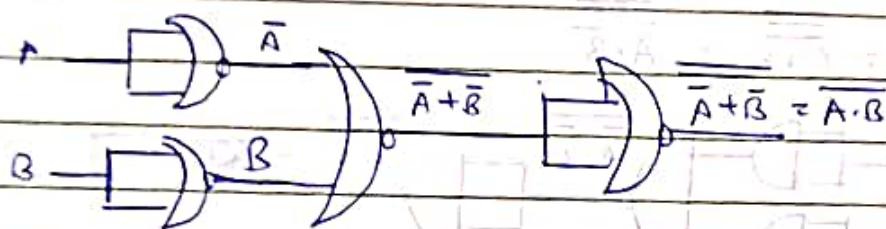


o3 AND  $y = \overline{A \cdot B} + \overline{\overline{A} + \overline{B}}$



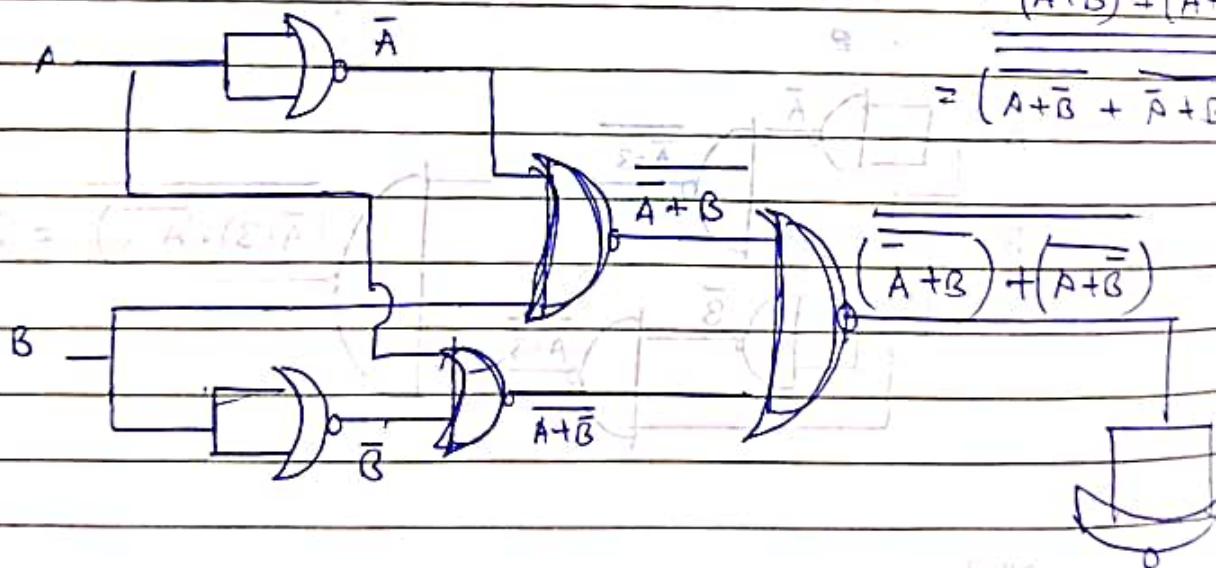
o5 NAND

$$y = \overline{A \cdot B} = \overline{\overline{A} + \overline{B}}$$



o5 EXOR

$$y = \overline{A \cdot B} + A \cdot \overline{B} = (\overline{A \cdot B}) \cdot (\overline{A \cdot \overline{B}}) = (\overline{A + \overline{B}}) \cdot (\overline{A + B})$$



$$\overline{A \cdot B} + A \cdot \overline{B} = (\overline{A + \overline{B}}) \cdot (\overline{A + B})$$

\* Reduce the expression,  $(C(A-BB-1)C)(A-B)A$

$$\begin{aligned}
 i) f &= A + B [AC + (B+C)D] \\
 &= A + BAC + BD(B+C) - 1 + B \\
 &= A + BAC + BD(B+1-C) - 1 \\
 &\Rightarrow A + BAC + BD + B - BD - BDC \\
 &= A + BAC + BD(1-C) \\
 &= A + BAC + BD - BDC \\
 &= A(1+BC) + BD(1-C) \\
 &= A + BD
 \end{aligned}$$

$$\begin{aligned}
 ii) f &= (B+BC) (B+\bar{B}C)(B+D) \\
 &= [B(1+C)] (B+(1-B)C)(B+D) \\
 &= B(B+C-BC)(B+D) \\
 &= B(B+BC-BC)(B+D) \\
 &= B(B+D) \\
 &= B + BD \\
 &= B(1+D) \Rightarrow B //
 \end{aligned}$$

$$iii) A [B+\bar{C} (\overline{AB} + \bar{A}C)]$$

$$\begin{aligned}
 f &= A [B+\bar{C} (\overline{AB} + \bar{A}C)] \\
 &= A [B+(1-C)(1-(AB+\bar{A}C))] \\
 &= A [B+(1-C)(1-\bar{A}B - (1-\bar{A})C)] \\
 &= A [B+(1-C)(1-\bar{A}B - C + AC)] \\
 &= A [B+(1-C)(1-AB-AC)] \\
 &\Rightarrow A [B+(1-C)(\cancel{1-AB-AC})] \\
 &= A (B + (1-C)(-AB - C))
 \end{aligned}$$

$$\begin{aligned} A \{ B + (1 - c)(1 - AB - AC) \} \\ = A \{ B + 1 - c - AB - AC - BC + ABC \} \\ = A \{ B + 1 - c - AB \} \\ = A \{ 1 - c - AB \} \\ = A(1 - c - AB) \\ = A(1 - c) + A \\ = cA + A \end{aligned}$$

$$\begin{aligned} A \{ B + (1 - c)(1 - AB - c) \} \\ = A \{ B + (ABC + c - AB - c) \} \\ = A \{ B + ABC + AB \} \\ = AB + ABC + AB \end{aligned}$$

Method 1  $AB$

$A + \bar{B}R$   
 $\text{Method 2}$   
 $A + B + \bar{B}R$

## Tricks

Method 1

$$A + 1 = A + (1 + B)$$

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

Manipulation

$$= A(1 + B) + \bar{B}R$$

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

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

(Extra) L10 & L12 E&A

Method 2 Distributive property

$$A + \overline{A}B$$

$$(A + \overline{A}) \cdot (A + B)$$

$$A + B$$

Method 3 logic gate

$$Y = A + \overline{A}B$$

$$\text{No. of combination} = 2^2 = 4$$

$$A \quad B \quad Y$$

$$0 \quad 0 \quad 0$$

$$1 \quad 0 \quad 1 + 0 \cdot 0 = 1$$

$$0 \quad 1 \quad 1$$

$$1 \quad 1 \quad 1$$

OR GATE

$A + B$

$$Q) A \cdot \overline{B} \cdot C + \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C} + A \overline{B} \cdot \overline{C}$$

$$\overline{B} \cdot (AC + \overline{A}C + \overline{A}\overline{C} + A\overline{C})$$

$$\overline{B} \cdot (C(A + \overline{A}) + \overline{C}(\overline{A} + A))$$

$$\overline{B} \cdot (C + \overline{C})$$

$\overline{B}$

$$Q) A \cdot C + A \cdot D + A \cdot \overline{C} \cdot \overline{D} + B \cdot R$$

$$A(C + D + \overline{C} \cdot \overline{D} + B)$$

$$A(C + D + \overline{C} \cdot \overline{D} + B)$$

$$A(1 + B) = A$$

AND logic  
OR logic  
NOT logic } Basic logic / Basic gates

Gates  $\rightarrow$  Basic building blocks of ckt.

EXOR

EXNOR

NAND

NOR

Universal

Property

Derived gates

{ Two logics  
viz AND, OR  
universal

$$\text{NOR} \equiv \text{NOT(OR)}$$

$$\text{NAND} \equiv \text{NOT(AND)}$$

gates like

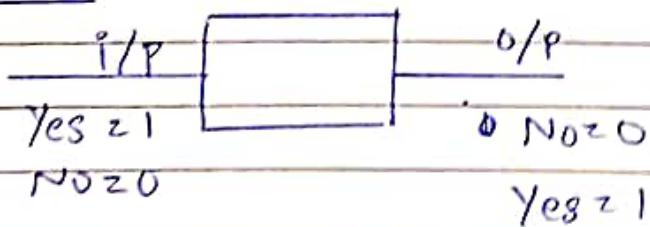
$$\text{EXOR} \equiv \bar{A}B + A\bar{B}$$

as we can  
create all

$$\left\{ \begin{array}{l} (\text{AND}(\text{NOT}(A), B)) \text{ OR} \\ (\text{AND}(A, \text{NOT}(B), A)) \end{array} \right\}$$

gates from them }

NOT



eg,

$$AB + (A+B)(A+C)$$

$$A \cdot A + BC + A \cdot B + A \cdot C$$

$$A + BC + \overline{A}(B+C)$$

~~$$A[1+B+C] + BC$$~~

$$A \oplus BC //$$

$$A + BC + AB + AC = \bar{A} + \bar{B}A = Y \quad (1)$$

$$A(1+B) + BC + AC$$

$$A + BC + AC$$

$$A(1+C) + BC$$

$$A + BC$$

(+) (P)  $(A+B) \cdot (B+C)$  using 2nd commutivity

same method

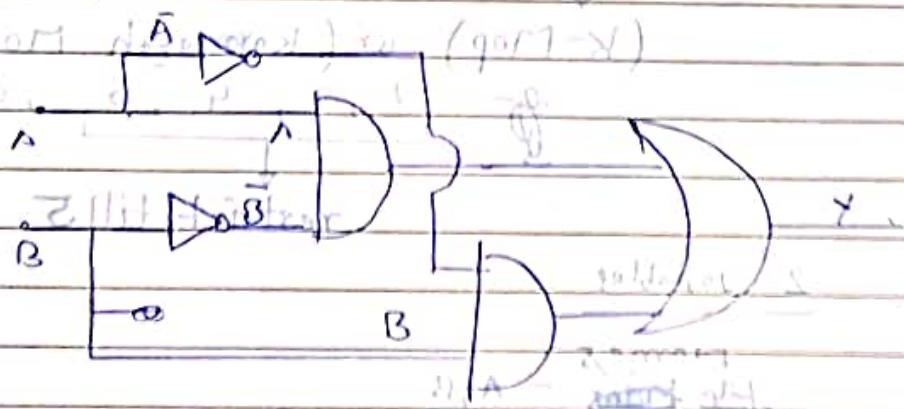
$$= AC + B$$

$$\bar{A} + 0 \cdot \bar{B}(\bar{A} + B) + \bar{B}\bar{A} + \bar{B}A = Y$$

(+) (P) Implement a clk +  $\bar{B}A + \bar{B}A =$

$$\bar{B}Y = \bar{A}B + A\bar{B} + \bar{B} \text{ using basic gate}$$

~~negation function with feedback loop~~



following implementation into logic

(P)  $Y = ABC + A\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C$

+ correction  $A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}, (\bar{A} + \bar{B}) + \bar{A}\bar{B}C$

$$ABC + \bar{B}\bar{C} + A\bar{B}C$$

$$AB + \bar{B} + B(\bar{C} + AC)$$

$$\bar{C}(AB + \bar{B}C) + A\bar{B}C \quad (1-B)C$$

$$A(\bar{B}C + \bar{B}C) + \bar{B}\bar{C} \quad -ABC + A\bar{B}C + \bar{B}\bar{C}$$

$$A(B(1-C) + \bar{B}C) + \bar{B}\bar{C} \rightarrow A(B-C + \bar{B}C) + \bar{B}\bar{C}$$

HW

$$\text{Q) } Y = AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC$$



AND = 1  
 OR = 1  
 NOT = 1

NOT = 3  
 OR = 3  
 AND = 8

② (Minimize the given equation with Boolean logic)

$AB\bar{C}$

$$\begin{aligned}
 Y &= AB\bar{C} + A\bar{B}C + (A+\bar{A})\bar{B}\bar{C} \\
 &= AB\bar{C} + A\bar{B}C + \bar{B}\bar{C} \\
 &= AB - ABC + AC - ABC + \bar{B}\bar{C}
 \end{aligned}$$

### Graphical Method for Circuit Minimization

↓  
(K-Map) or (Karnaugh Map)

2, 3, 4, 5, etc., variables ↓  
so

restrict till 5

2 variables

names

~~like terms~~ = A, B

and are independent variables

$$2\bar{A} + 2\bar{B} + 2\bar{A}\bar{B} + 2A = Y(5)$$

In case of (4 binary) no. of combination = 4

No of combinations =  $2^{(\text{no of variables})}$

$$(2+2)(2+2) + 2\bar{A}\bar{B} + 2A =$$

$$2\bar{A} + (2\bar{B} + 2A) =$$

$$2\bar{A} + (2\bar{B} + 2A) = 2\bar{A} + (2\bar{B} + 2A) =$$

$$2\bar{A} + (2\bar{B} + 2A) = 2\bar{A} + (2\bar{B} + (2-1)A) =$$

$(101110)_2$  ↗ lowest significant bit (LSB)

↙ Most  
significant  
bit (MSB)

K-Map ↗ MSB

	A	B	0	1	0	1	0	1	0	1	0	1
0	0	0	0	1	0	1	0	1	0	1	0	1
1	0	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	2	1	0	1	0	1	0	1
1	1	0	1	0	3	1	1	0	1	1	0	1

cell

min term	Max term	Y
000	111	0
001	110	1
010	101	2
011	100	3
100	011	0
101	010	1
110	001	2
111	000	3

SOP  $\Rightarrow$  Sum of products (interested in logic 1)

POS  $\Rightarrow$  Product of sums (interested in logic 0)

$$Y = AB\bar{C} + (\bar{A}\bar{B}\bar{C}) + A\bar{B}\bar{C} + \bar{A}B\bar{C}$$

↓ -

$$110 \quad 000 = 10101$$

karimpor ali fid S o/(E) tresser of  
mangalgarh 10/11/2019 4.5

A B C ↓

De eq.				Product term SOP	Product term Minterm	$\Sigma m(0, 4, 5, 6)$	Y = $\sum m(0, 4, 5, 6)$
	A	B	C				
✓ 0	0	0	0	$ABC$	1	1	$\Sigma m(0, 4, 5, 6)$
1	0	0	1	$\bar{A}B\bar{C}$	0	0	or SOP
2	0	1	0	$\bar{A}B\bar{C}$	0	0	
3	0	1	1	$\bar{A}BC$	0	0	
✓ 4	1	0	0	$A\bar{B}\bar{C}$	1	1	
✓ 5	1	0	1	$A\bar{B}C$	1	1	
✓ 6	1	1	0	$AB\bar{C}$	1	1	
7	1	1	1	$ABC$	0	0	

$$y = \sum m(0, 4, 5, 6) \rightarrow \text{SOP minterm representation}$$


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

(min term)  $\rightarrow$  standard form  $\rightarrow$  SOP

$$\Sigma \bar{A}y = \sum m(1, 3) + 3BA = Y$$

10 max value  $\rightarrow (3)_{10} = 011$

To represent  $(3)_{10}$  2 bit is required  
 i.e. 2 variable problem

Say

$$A \Rightarrow \text{MSB}$$

$$B \Rightarrow \text{LSB}$$

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

If in group there is single logic 1 then it is always true value

Groups of logic - 1

Validity of groups

$$\text{No of logic 1's} = \frac{\text{No of variable changes per row}}{2}$$

variables

$$\therefore Y = B$$

$$\text{SOP eqn: } Y = \overline{AB} + AB$$

$$= B \cdot (\overline{A} + A)$$

$$\therefore Y = B$$

beginning

$$\text{eg: } Y = \sum m(2, 3)$$

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

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

$$= A$$

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

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

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

$$\sum m(1, 2) = 2$$

$$AB + A\bar{B} = 4$$

Invalid

	AB	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	AB
	00	01	10	11	10
$\bar{C}$ 0	1	0	1	0	1
	0	2	2	0	0
$\bar{C}$ 1	0	0	0	1	0
	1	3	0	3	0

HW

$$\sum m(0, 4, 5, 6)$$

ab + cd = 0000, 0100, 1100, 1000

0000 01 11 10

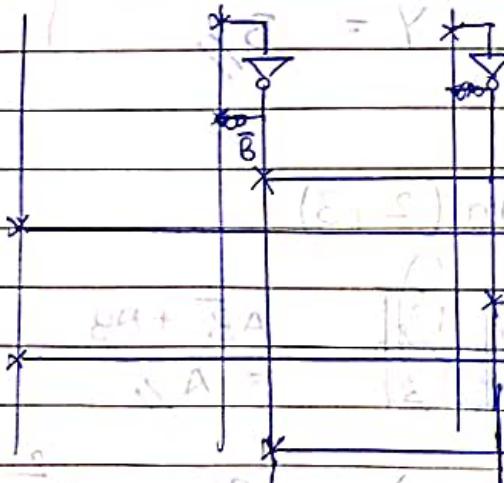
$\bar{C}$ 0	1	0	1	1
$\bar{C}$ 1	0	0	0	1

1) cheek  
2) mouth  
3) group

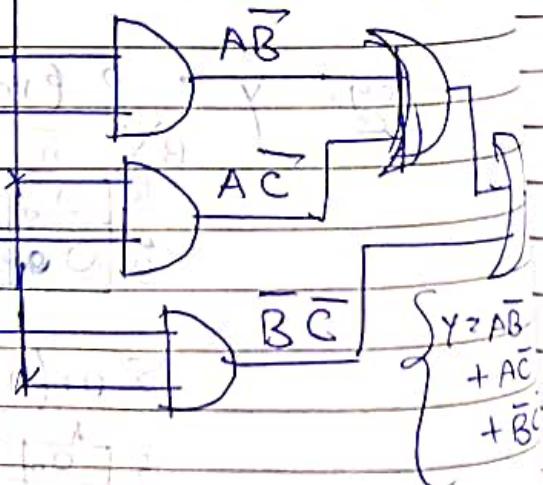
$$y_1 = A\bar{B} + \bar{A}\bar{C} + \bar{B}\bar{C}$$

$$y = y_1 + y_2 + y_3$$

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



Minimized  
ckt implement



$$\left\{ \begin{array}{l} Y = AB \\ + AC \\ + BC \end{array} \right.$$

QUESTION

HW

$$Y = \sum m(0, 1, 3, 5, 7)$$

$$Y = \sum m(0, 3, 5, 7)$$

$$Y = \sum m(1, 3, 7)$$

$$Y = \sum m(0, 1, 2, 4)$$

1)

AB	00	01	11	10
C	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
0 $\bar{C}$	(1) 0	0 2	0 6	0 4
1 C	(1) 1	(1) 3 (1) 7 (1) 5		

$$Y_1 = AC$$

$$Y_2 = BC$$

$$Y_3 = \bar{A}C$$

$$Y_4 = \bar{A}\bar{B}$$

$$Y_5 = \bar{B}C$$

$$Y = Y_1 + Y_2 + Y_3 + Y_4 + Y_5$$

$$= AC + BC + \bar{A}C + \bar{A}\bar{B} + \bar{B}C$$

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

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

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

2)

AB	00	01	11	10
C	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
0 $\bar{C}$	0 0	0 2	0 6	0 4
1 C	0 1	(1) 3 (1) 7 (1) 5		

$$Y_1 = A\bar{C}$$

$$Y_2 = BC$$

$$Y = Y_1 + Y_2 = A\bar{C} + BC$$

3)

	$\bar{A}B$	$\bar{A}\bar{B}$	$\bar{B}B$	$AB$	$A\bar{B}$
$\bar{C}$	0, 0	0, 0	0, 0	0, 0	0, 0
C	1, 1	1, 3	1, 1	0, 0	0, 0

$$Y_1 = 0 \bar{A}C$$

$$Y_2 = BC$$

$$Y = Y_1 + Y_2$$

$$= \bar{A}C + BC = \bar{A}C + BC$$

4)

	$\bar{A}B$	$\bar{A}\bar{B}$	$AB$	$A\bar{B}$
$\bar{C}$	1, 0	1, 0	0, 0	0, 1
C	1, 0	0, 3	0, 7	0, 5

$$Y_1 = 0 \bar{A} \bar{C} = 0 \bar{A} \bar{C}$$

$$Y_2 = 0 \bar{B} = 0 \bar{B}$$

$$Y = Y_1 + Y_2 + Y_3$$

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

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

$$BA + A$$

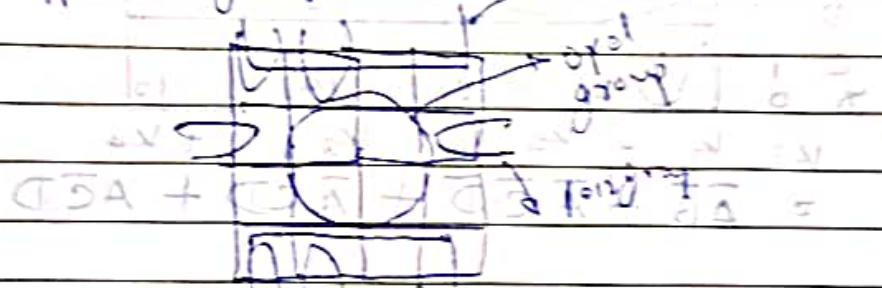
$$BA + A$$

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

\* Rules

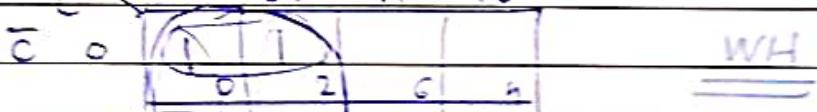
- \* No zeros allowed
- \* Groups can be vertical/horizontal never diagonal
- \* Overlapping allowed
- \* Group should be as large as possible
- \* Group must contain  $2^n$  cells

Type of group



$$\sum m(0, 1, 2, 3) = F_{AB}$$

$$\sum m(0, 1, 2, 3, 5) = \overline{B}A + \overline{B}C + BC$$



$$\sum m(0, 1, 2, 3) = F_{AB}$$

$$(E_1, E_2, E_3, P, Z) \leftarrow (E_1, E_2, E_3, P, Z) \oplus (\overline{A} + \overline{B}C)$$

$$(E_1, E_2, \overline{A}\overline{B} + \overline{A}B + \overline{B}\overline{A}C + \overline{A}C + \overline{B}C) = E_1, E_2, \overline{B}C$$

$$(E_1, E_2, \overline{A} + \overline{B}C, \overline{B}C) = E_1, E_2, \overline{B}C$$

$$(E_1, E_2, \overline{B}C, \overline{B}C) = E_1, E_2, \overline{B}C$$

4 variable K-maps

		AB	$\bar{A}\bar{B}$	$\bar{A}B$	$AB$	$y = \sum m(0, 1, 2, 3, 4, 7)$
		CD	00	01	11	10
C	0	V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	V <sub>4</sub>	
	1					
D	0		1	1	1	1
	1					
C	1		1	1	1	1
	0					
D	1		1	1	1	1
	0					

$V_1 = \bar{A}\bar{B}\bar{C}\bar{D}$   
 $V_2 = \bar{A}\bar{B}CD$   
 $V_3 = A\bar{B}\bar{C}\bar{D}$   
 $V_4 = A\bar{B}CD$   
 $y = V_1 + V_2 + V_3 + V_4$   
 $= \bar{A}\bar{B} + \bar{A}\cdot\bar{C}\cdot\bar{D} + \bar{A}\cdot C\cdot D + A\bar{C}\cdot D$

NOT = 4

OR = 3

AND = 7

} for 2  
input

HW

1)  $y = \sum m(0, 1, 3, 5, 7, 8, 9, 11, 13, 15)$

2)  $y = \sum m(0, 1, 8, 9)$

3)  $y = \sum m(8, 9, 10, 13, 15)$

4)  $y = \sum m(1, 2, 3, 6, 7, 9, 10, 11, 13)$

5)  $y = \sum m(0, 9, 13, 15)$

6)  $y = \sum m(1, 2, 3, 13, 14, 15)$

HW

1)

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

$$Y = \overline{ABCD} + \overline{B} \bar{C} \bar{D} + \overline{ABC} + \bar{D}$$

2)

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

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

 $\bar{C} \bar{D} A + \bar{C} D \bar{A}$ 

3)

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

$$\begin{aligned} & Y = \bar{A} \bar{B} \bar{C} \\ & + \bar{A} B D \\ & + A \bar{C} \bar{B} \\ & + A \bar{C} D \end{aligned}$$

4)

		$\bar{A}B$	$\bar{A}\bar{B}$	$\bar{A}B$	$AB$	$A\bar{B}$
		CD	00	01	11	10
$\bar{C}$	0		0	4	12	8
	1	D		5	13	19
$\bar{D}$	1					
	0	C	7	1	15	11
$\bar{D}$	1	D	3	7	15	11
	0	C	11	12	16	14
$\bar{D}$	1	D	0			
	0					

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

5)

		$\bar{A}B$	$\bar{A}\bar{B}$	$\bar{A}B$	$AB$	$A\bar{B}$
		CD	00	01	11	10
$\bar{C}$	0		1	4	12	8
	1			5	13	19
$\bar{D}$	1	C	7	1	15	11
	0	D	3	7	15	11
$\bar{D}$	1	C	11	2	6	14
	0	D	0			

$$y = \bar{A}\bar{B}\bar{C}\bar{D} + ABD + \cancel{\bar{A}\bar{B}\bar{C}\bar{D}} + \bar{A}\bar{C}D$$

6)

		$\bar{A}B$	$\bar{A}\bar{B}$	$\bar{A}B$	$AB$	$A\bar{B}$
		CD	00	01	11	10
$\bar{C}$	0		0	4	12	8
	1			5	13	9
$\bar{D}$	1	C	7	1	15	11
	0	D	3	7	15	11
$\bar{D}$	1	C	11	2	6	14
	0	D	0			

$$y = \bar{A}\bar{B}D + ABD + \cancel{\bar{A}\bar{B}\bar{C}\bar{D}} + \bar{A}\bar{C}D$$

Euclidean k-Map

- 5 var

- GR only

~~180~~  
~~220~~

$$F_{\text{min}} = \overline{C} \overline{D} \overline{E} + \overline{C} \overline{D} E + \overline{C} D \overline{E} + C \overline{D} \overline{E}$$

$$+ \overline{C} \overline{D} E + C D \overline{E}$$

min 0	0	1	2	3	4	5	6	7
min 1	0	1	2	3	4	5	6	7
min 2	0	1	2	3	4	5	6	7
min 3	0	1	2	3	4	5	6	7
min 4	0	1	2	3	4	5	6	7
min 5	0	1	2	3	4	5	6	7
min 6	0	1	2	3	4	5	6	7
min 7	0	1	2	3	4	5	6	7

~~ABC~~

					110	111	101	100	WH
(0)DE	000	001	011	0010	<del>000</del>	<del>000</del>	<del>001</del>	<del>000</del>	<del>000</del>
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	0	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
0	2	6	14	10	26	30	22	18	



All k-maps have  
folowing property

$$y = \sum m(0, 1, 3, 8, 9, 10, 13, 19, 21, 23, 29, 31)$$

ABC		000	001	011	010	110	111	101	100
DE		00	01	12	7	24	28	20	16
00	1	4	13	19	25	1, 29	1, 21	17	
01	1	5	13	19	27	1, 31	1, 23	11	
12	1	1	15	11	24	80	21	10	
19	2	6	14	10	26				

$$y = ACE + \bar{B}\bar{C}DE + \bar{A}\bar{B}CD + \bar{A}\bar{B}\bar{C}\bar{D}$$

~~+ B̄C̄D̄~~

OR

we can club (28, 19) with (3, 7) } using told  
 (13, 9) with (29, 21) } property

and make 4 group which  
 is optimum

~~HW~~

$$1) y_1 = \sum m(0, 1, 3, 8, 9, 13, 19, 21, 23, 29)$$

$$2) y_2 = \sum m(0, 1, 2, 3, 4, 9, 10, 11, 13, 15, 17, 29, 31)$$

$$3) y_3 = \sum m(0, 2, 9, 13, 15, 17, 19, 21, 23, 27, 29, 31)$$

red open by HA  
 dragon writing

## Arithmetic Circuits

### Binary Addition Rules

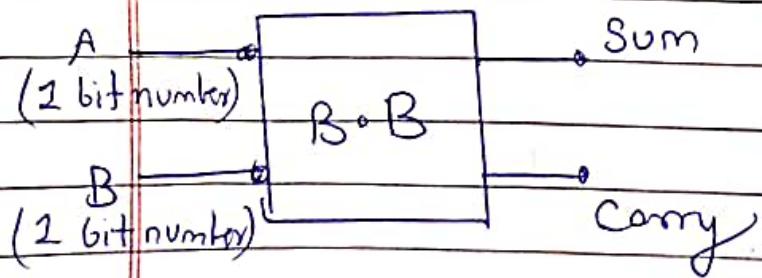
A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Sos  $\Rightarrow$  Concept

Th	H	T	U
2	3	8	
9	4	5	
1	1	7	3
+1			
8			

{Half addition  
with carry}  
Full addition  
with carry

### Half Adder



Say  $A = \text{MSB}$ ; A & B are both 1-bit numbers

I/P		O/P	
A	B	Carry Cout=0	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

For Sum (S)

		00	01
		00	01
		00	01
B	00	0	0
		0	1
	01	1	0
			2
			3

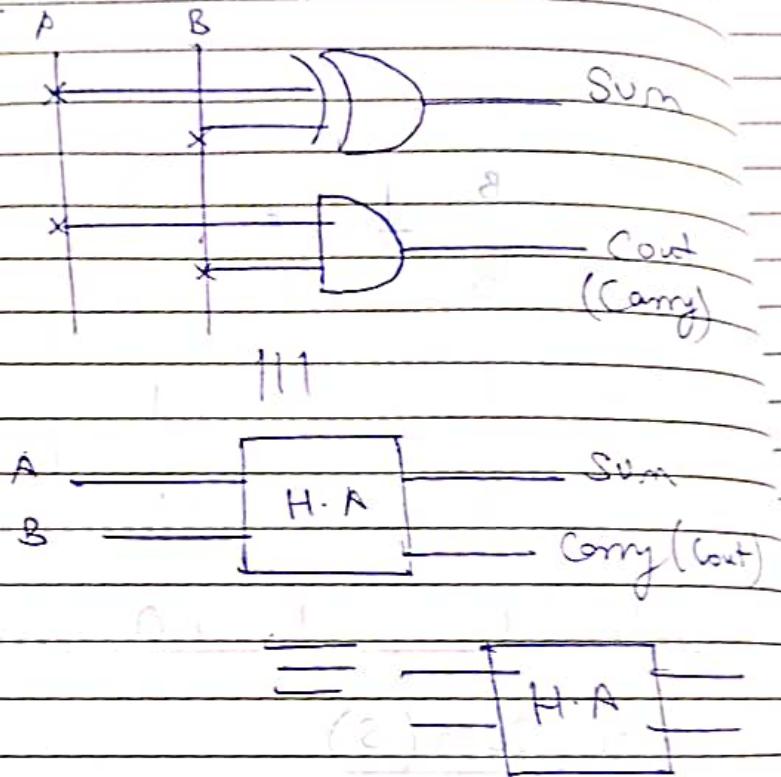
$$S = \overline{AB} + \overline{A}\overline{B} \Rightarrow A \oplus B$$

For Carry (Cout)

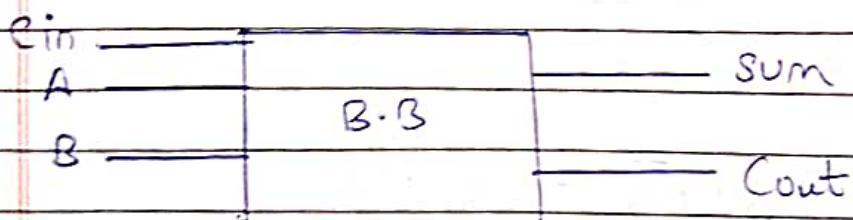
		0	1
		0	0
		0	0
B	0	0	0
		0	1
	1	0	1
			2
			3

$$Y = A \cdot B$$

### Realisation



### Full Adder



8

I/P			O/P	
A	B	Cin	Sum	Cout sum
0	0	0	0	0 0
1	0	0	1	0 1
2	0	1	0	0 1
3	0	1	1	0 1
4	1	0	0	1
5	1	0	1	0 1
6	1	1	0	0
7	1	1	1	1

AB		Sum //			
C	00	01	11	10	11
0	0	1	2	5	14
1	1	1	1	1	1
	1	1	3	7	5

$$\begin{aligned}
 y &= \overline{ABC} + \overline{B}\overline{C} + A\overline{BC} + A\overline{B}\overline{C} \\
 &\quad \boxed{\overline{A}(\overline{BC} + \overline{B}\overline{C}) + \overline{A}B(ABC + A\overline{BC})} \\
 &\quad \boxed{A(B + C) + A(BC + \overline{B}\overline{C})} \\
 &= A(B + C)
 \end{aligned}$$

Cout

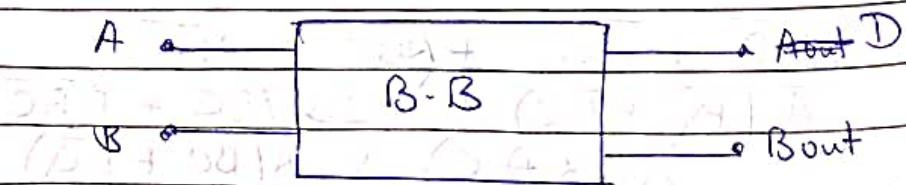
		AB	00	01	11	10
		C	0	1	1	0
0	0	0	2	3	4	5
	1	1	3	7	1	5

$$y_2 = \delta BC + AB + \overline{A} \overline{B} C$$

Binary Subtraction

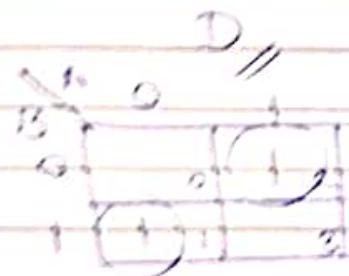
	A	B	Borrow	Difference	
0	0	0	0	0	
1	0	1	1	1	
1	1	0	0	1	
3	1	1	0	0	

Half Subtractor



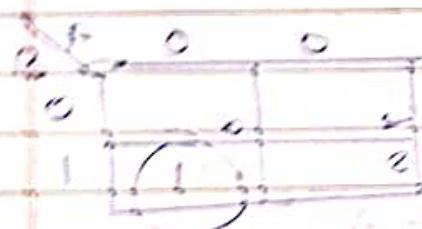
$$D = A \oplus B$$

$$B_{out} = \bar{A}B$$

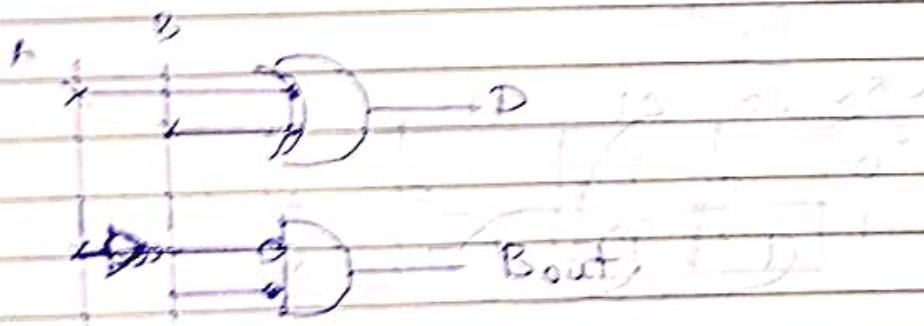


$$Y = AB + \bar{A}B = A \oplus B$$

Binary //



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



Full Subtractor  $+ \bar{A}B + CA = Z$

	I/P		O/P		
	A	B	Bin	Bout	D
0	0	0	0	0	0
1	0	0	1	1	1
2	00	1	0	1	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	0	⑥ b
6	1	1	0	0	0
7	1	1	1	1	1

	D <sub>1</sub>		D <sub>2</sub>		D <sub>3</sub>	
	AB	00	01	11	10	
C	0	0	1	2	6	4
1	1	1	3	1	7	5

$$y_1 = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}\bar{C}$$

	Bout		B <sub>3</sub>		B <sub>2</sub>		B <sub>1</sub>		B <sub>0</sub>	
	AB	00	01	11	10					
C	0	0	1	2	6	4				
1	1	1	3	1	7	5				

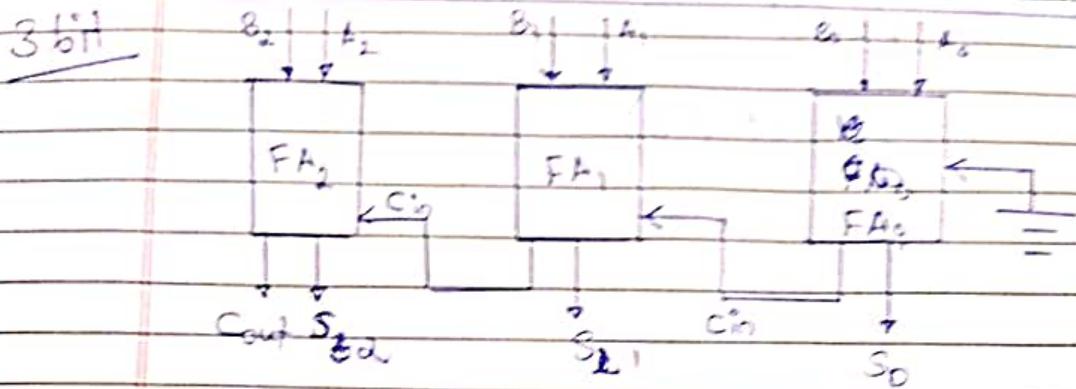
$$y_2 = AC + \bar{B}\bar{A}B + B\bar{e}$$

B<sub>0</sub>

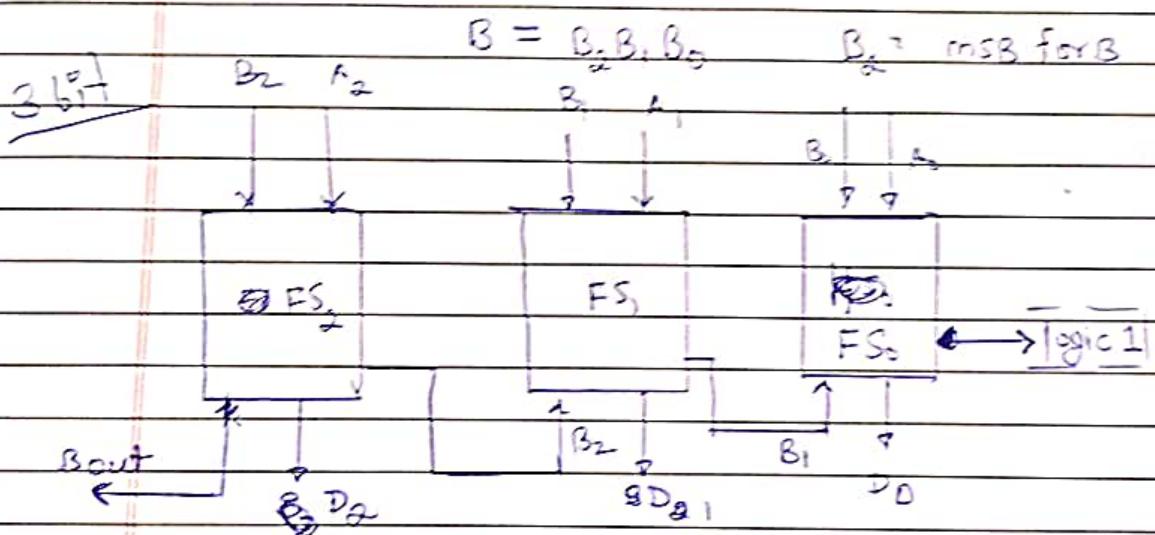
B<sub>1</sub>

C<sub>0</sub>

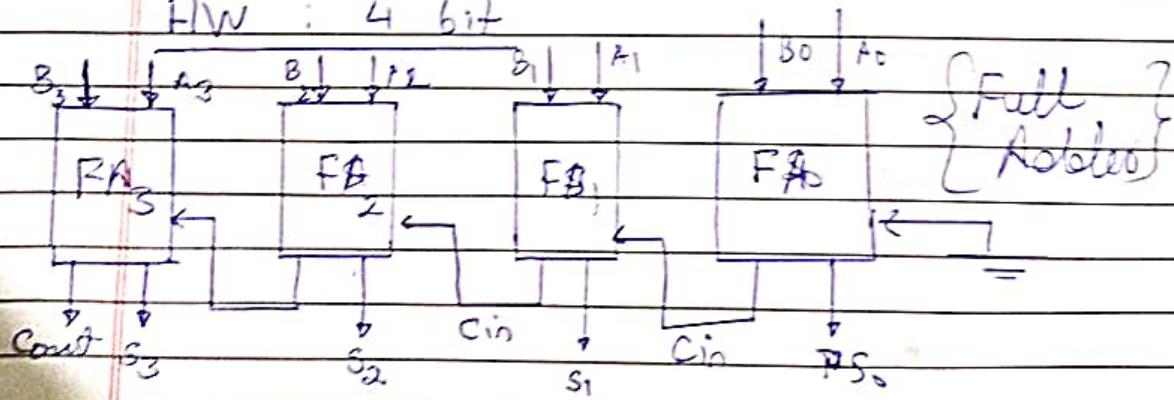
4 bit Parallel Adder & 4-bit Parallel Subtractor



where  $A_4 = A_3, A_2, A_1$        $A_2 = \text{msb for } A$



HW : 4 bit



### Q Complement

$$x - y = x + (-y)$$

$$x - y = x + \Xi$$

QUESTION  
ANSWER

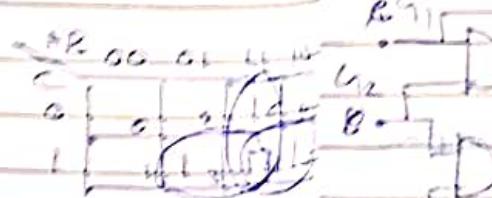
P) A family having 3 generations hold a locker in a bank. To open the locker the condition of the bank is that atleast 2 generations must be present in the bank along with their respective keys. Design a digital system for this locker.

	R	C	G	B	Y = FAB	B, C / G	Z
0	0	0	0	0	0	0	0
1	1	0	0	0	1	1	1
2	0	1	0	0	0	0	0
3	0	0	1	0	0	0	0
4	1	0	0	0	1	1	1
5	0	1	1	0	1	1	1
6	1	1	0	0	1	1	1
7	1	1	1	0	1	1	1

It is expected to design a locker upto 5 yrs.  
The condition of bank is that atleast 2  
generations must be present in bank along with their keys.

There are 3 generation of family having their  
respective keys

Normal clock will be  
A(G<sub>1</sub>, G<sub>2</sub>, G<sub>3</sub>) address  
generations respectively



$$Y = BC + AB + B$$

If any generation person available in bank with key = 1 else with 0.

$$Y = G_1 G_2 + G_1 G_3$$

Soln

$C_1, C_2, C_3, Y$

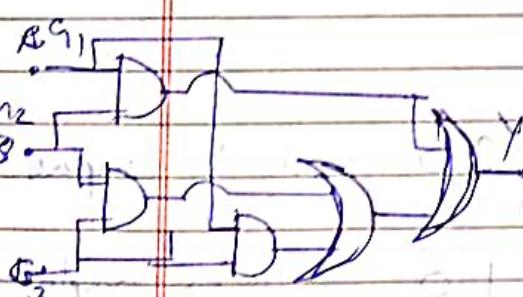
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

ab locker gets opened in hundreds  
it by 2 else with 0.

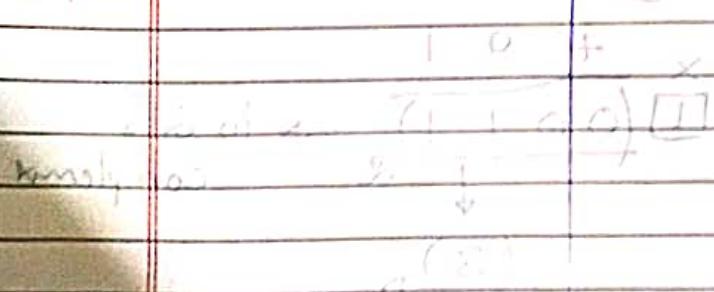
∴ Truth Table is as follows -

$$\left\{ \begin{array}{l} Y = \\ \quad C_2 C_3 + C_1 C_2 \\ \quad + C_1 C_3 \end{array} \right\}$$

$C_3$	0	1	2	3	4
$C_2$	0	1	0	1	0
$C_1$	1	1	1	0	1
	0	1	0	1	0



$$AB + AC$$



Q)  $\frac{(-7)}{D} = ? = (2)$  in 9's complement

~~(1)~~

= 3<sub>D</sub> in 10's complement

~~Q) (800)<sub>10</sub> in 9's comp~~  
~~999~~  
~~- 800~~  
~~199 ✓~~

~~eg/~~ 
$$\begin{array}{r} 9 \\ -7 \\ \hline (2)_{10} \end{array} = 9 + (-7)$$
  

$$= 0 + 2 \text{ in 9's complement}$$
  

$$= 2$$

~~(-16) 9's comp  
9-6=3 → 16  
2's comp~~

$$\begin{array}{r} 9 \\ -6 \\ \hline 3 \end{array} [9 + (-6)] = 4 \text{ in 10's complement}$$
  

$$9+4 \times 3 \text{ c.s}$$

~~sum = 3 //~~

~~A)  $\frac{(-3)}{7} = (4)$  in 7's complement  
 $\frac{(-3)}{2} (5)_{10} \text{ in } 8^{\text{th}} \text{ 1's complement}$~~

~~eg/~~ 
$$\begin{array}{r} (1010)_2 \\ - (0111)_2 \\ \hline \end{array} = ?$$

$$\begin{array}{r} - 0111 \rightarrow 1000 \text{ in 1's comp} \\ \hline 1010 \end{array}$$

~~10010 → in 1's comp~~

$$\begin{array}{r} \times + 9 \\ \hline \overbrace{(0011)}_2 \end{array} \rightarrow \text{in 2's complement}$$

$(3)_D$

00111



1000 → in 1's complement



1001 → in 2's complement

1010

$$\boxed{1} \underline{(0011)_2} \rightarrow 10(3)D$$

← ignored

MSB →  $\begin{cases} 0 & \text{+ve} \\ 1 & \text{-ve} \end{cases}$

4 bits = 1 nibble

8 bits = 1 Byte

2B = 1 word

2W = 1 double word

2 dW = 1 quadword

sign bit

$$8\text{-bit} \rightarrow \begin{array}{r} 0000\ 0000 \\ +1250 = 0 \\ 1111\ 1111 \end{array}$$

(-127 to 127)

for sign bit form

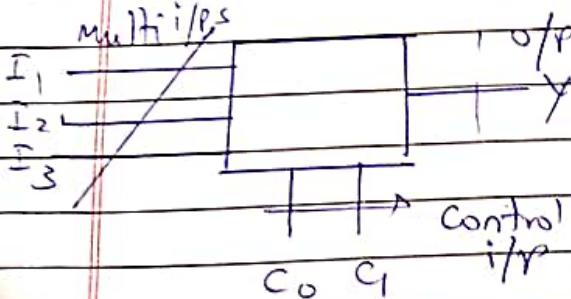
### Multiplexer (MUX)

\* Multiple i/p's

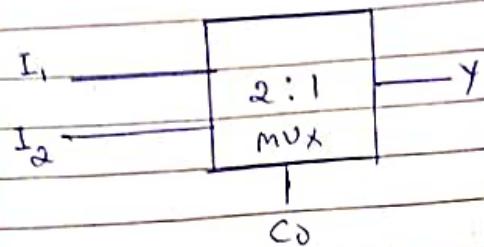
\* Only 1 o/p

\* # of i/p's =  $2^x$

\*  $x$  = no. of control signals



$2^x : 1 \Rightarrow 2^1 : 1 \rightarrow 2 : 1 \text{ MUX}$



C0	00	01	10	11
0	0	1	1	0
1	1	0	0	1

$$I_1 \bar{I}_2 \bar{C}_0 + \bar{C}_0 \bar{I}_1 I_2$$

$$\bar{C}_0 I_0 \bar{I}_1$$

$$0 \ 1 \ 0$$

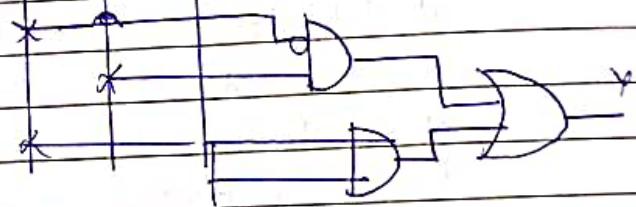
$$1 \ 0 \ 1$$

$$g) C_0 = 0 \quad y = I_0$$

$$C_0 = 1 \quad y = I_1$$

$$y = \frac{C_0}{C_0} I_0 + \frac{C_0}{C_0} I_1$$

$$C_0 \ I_0 \ I_1$$



(F2I + F2I -)

next file on HW

Design MUX

c) 4:1

b) 8:1

c) 16:1

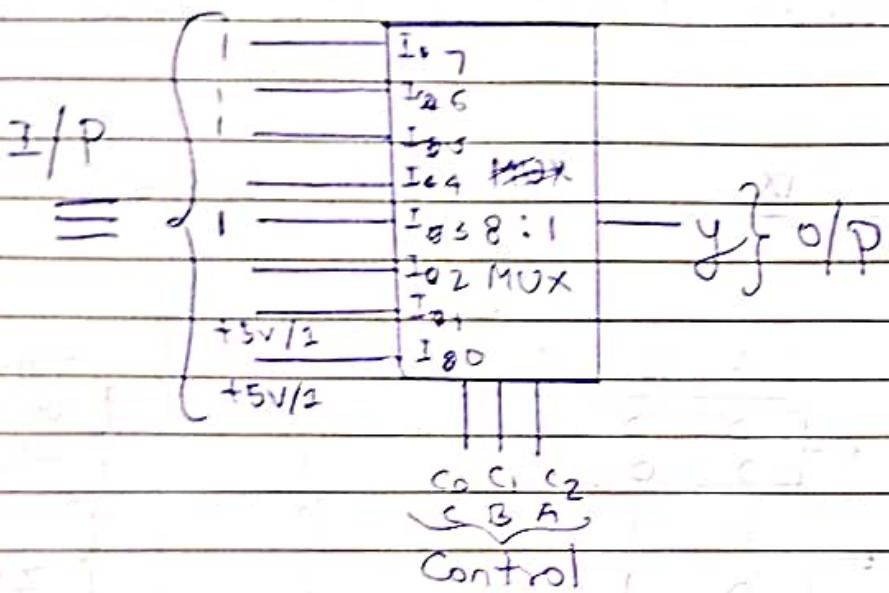
d) 32:1

e) 64:1

f) 5:1

so what do we do

8.1 MUX



Eg //

SOP

$$Y = \sum m(0, 1, 3, 5, 6, 7)$$

say  $c_2, c_1, c_0$  are iffs

$\Rightarrow$  MSB

(A)

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

$$ABC = 000 \quad z_1 \rightarrow 1 \quad Y = I_0 z_1$$

$$ABC = 001 \quad z_2 \rightarrow 1 \quad Y = I_1 z_1$$

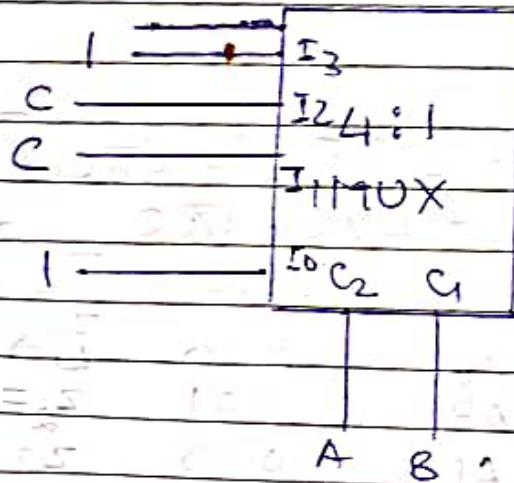
$$ABC = 010 \quad z_2 \rightarrow 0 \quad Y = I_2 z_2$$

HW

Solve the HW problems given for 3 variable, 4 variable & 5 variable k-maps by using proper mux.

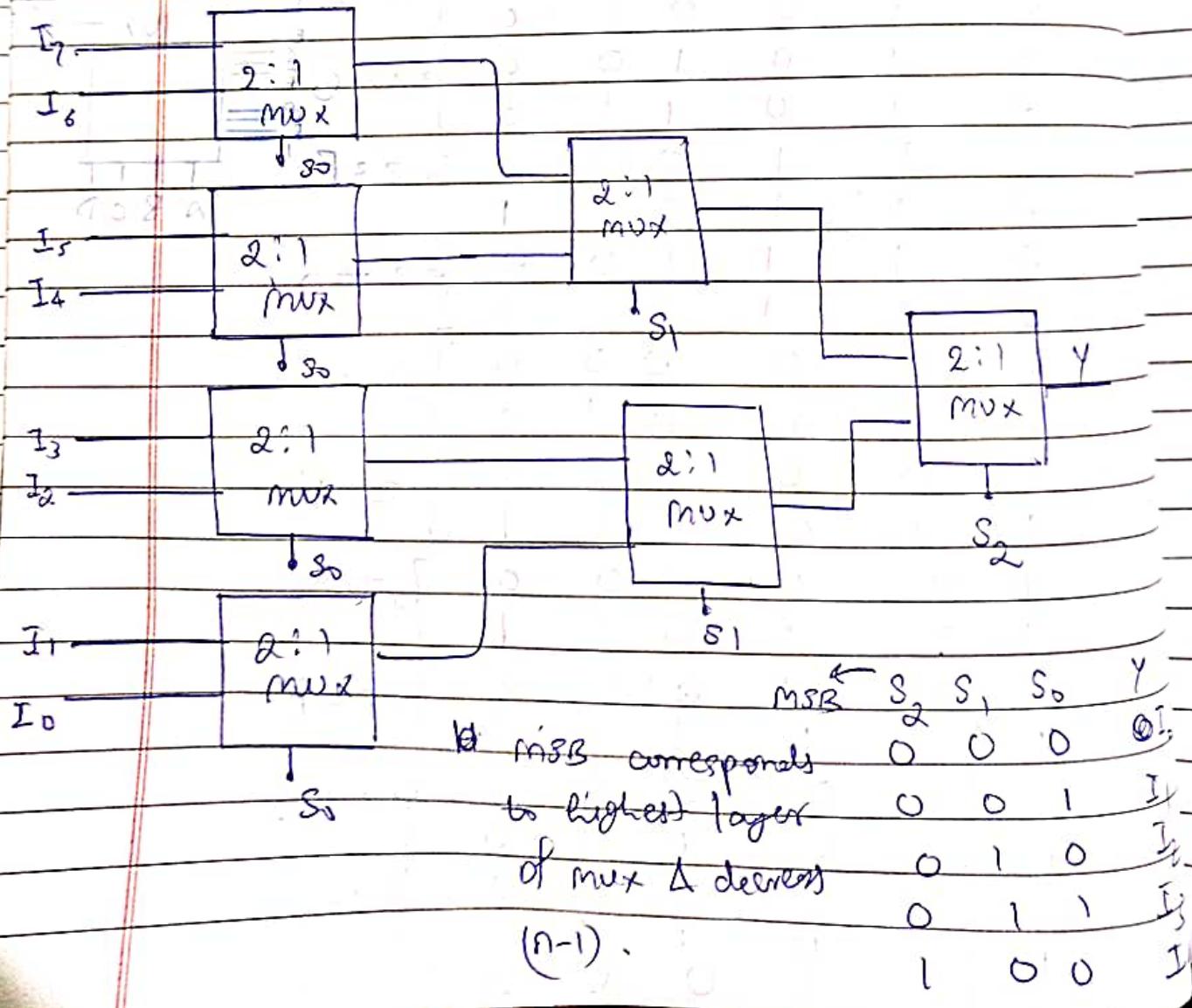
4:1 MUX $m I(0, 1, 3, 5, 6, 7)$ 

	A	B	C	Z	in respect of C
const. ←	0	0	0	1	} ab c } Z=1
	0	0	1	1	
const ←	0	1	0	0	} Z=C
	0	1	1	1	
const ←	1	0	0	0	} Z=C
	1	0	1	1	
const ←	1	1	0	1	} Z=1 } in respect of C
	1	1	1	1	



Design 8:1 MUX using 2:1 MUX only

<u>Desired MUX</u>	<u>Available MUX</u>
8:1 MUX	2:1 MUX
$S \rightarrow i/p$	$2^+ 1/p$
$I \rightarrow o/p$	$1 \rightarrow o/p$
$3 \rightarrow$ control signals	$1 +$ control signal



HW

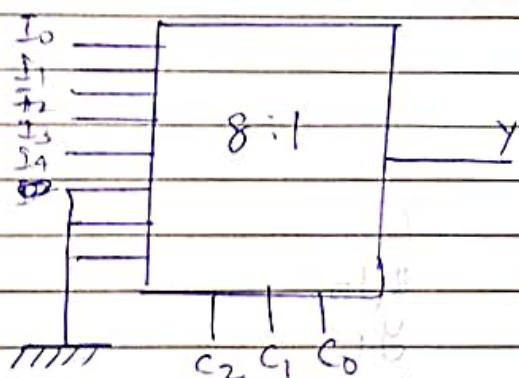
$$\left\{ \begin{array}{l} 16:1 \text{ using } 2:1 \\ 16:1 \text{ using } 2:1 \text{ & } 4:1 \\ 128:1 \text{ using } 8:1 \end{array} \right.$$

128

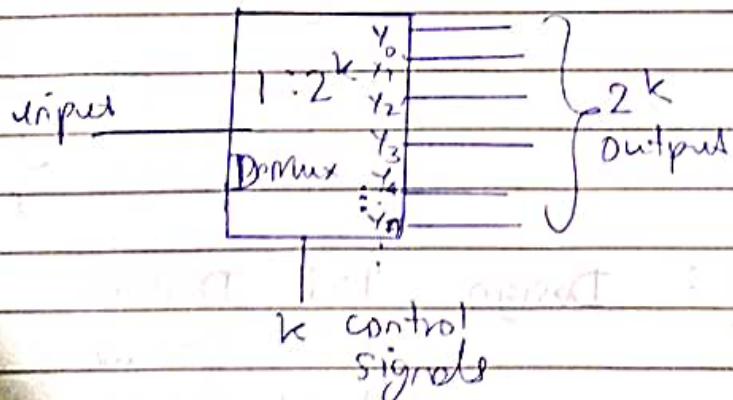
13:1 MUX

5:1 MUX

14:1 MUX  $\rightarrow 14 = 8 + 4 + 2$ 

$$\begin{matrix} & & \\ & \downarrow & \downarrow \\ 8:1 & \text{mux} & \text{mux} \\ \text{mux} & & \end{matrix}$$
DEMUX

No of P/p = 1

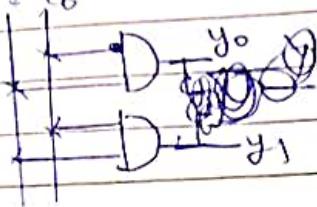
No of O/p =  $2^k$ where  $k = \text{no of control signals}$ k control  
signals

DEMUX

1:2 Demux

$$Y_0 = \bar{C}_0 I$$

$$Y_1 = C_0 I$$



1:4 Demux

X	S0	C1	C0	Y
0	0			Y0
0	1			Y1
1	0			Y2
1	1			Y3

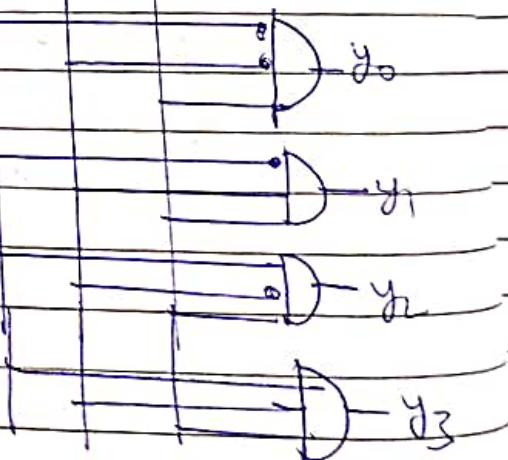
$$Y_0 = \bar{C}_1 \bar{C}_0 I$$

$$Y_1 = \bar{C}_0 C_1 C_0 I$$

$$Y_2 = C_1 \bar{C}_0 I$$

$$Y_3 = C_1 C_0 I$$

C1    C0    I



HW : Design 1:8 Demux

1:16 Demux

1:32 Demux

Note : MSB corresponds to ...

HW Design 1:8 Demux using 1:2 Demux  
only

1:4 using 1:2 only

1:64 using combination of  
1:2 & 1:4

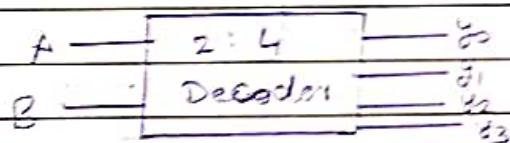
1:8 demux  
only

1:5 Demux

1:14 Demux

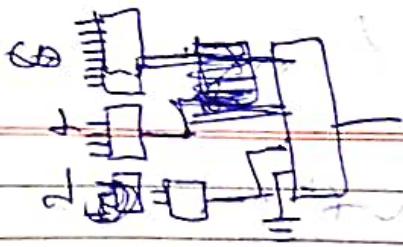
1:13 Demux

### DeCoder (2:4 Decoder)



I/O		O/P			
A	B	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$\begin{aligned}
 y_0 &= \bar{A}\bar{B} \\
 y_1 &= \bar{A}B \\
 y_2 &= A\bar{B} \\
 y_3 &= AB
 \end{aligned}
 \quad \left. \begin{array}{l} \text{Same as} \\ 1:4 \text{ Demux} \end{array} \right\}$$



HW :

3 : 8 Decoder

4 : 16 Demux

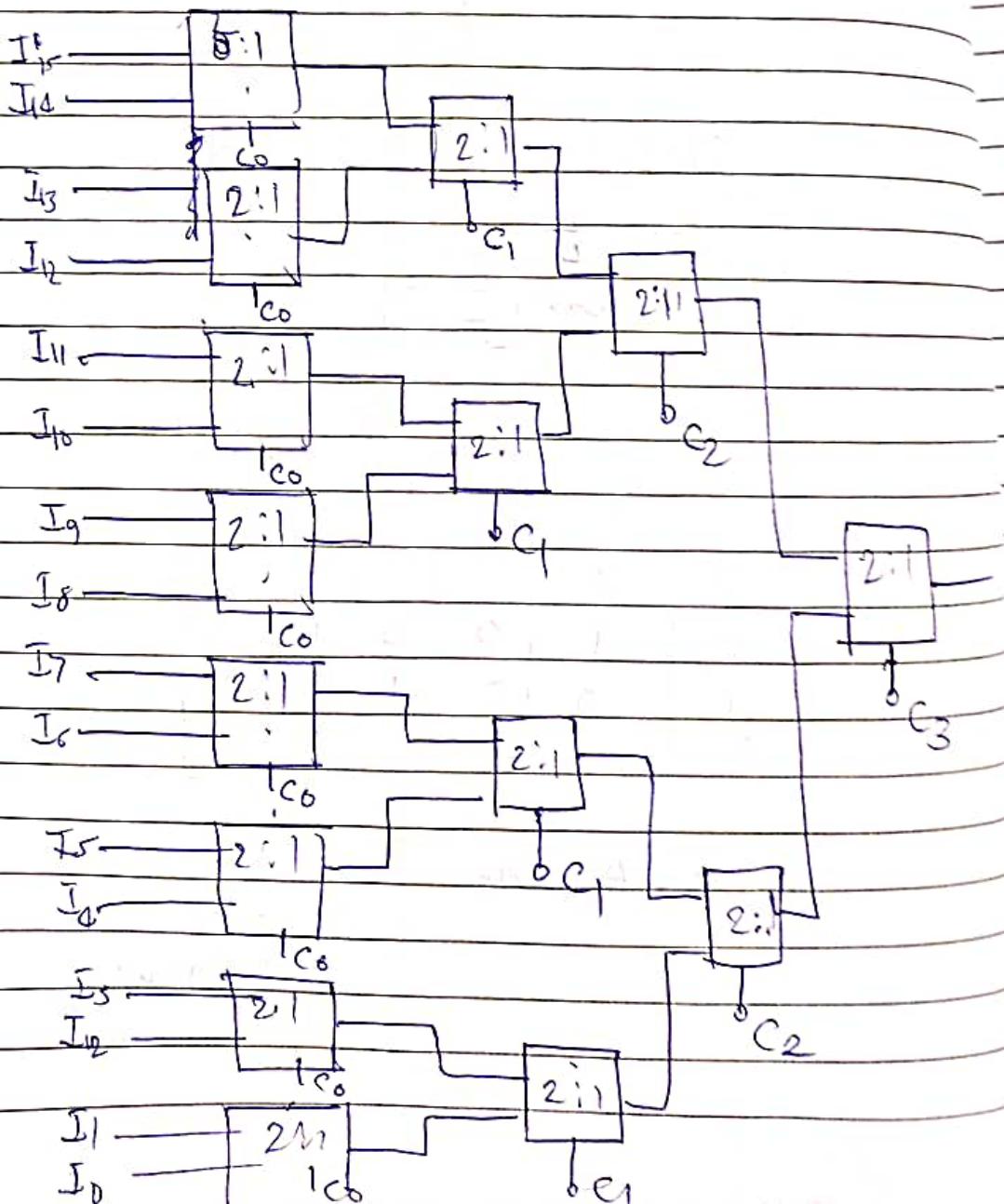
1 : 2 Decoder

5 : 32 Decoder { Done } core }

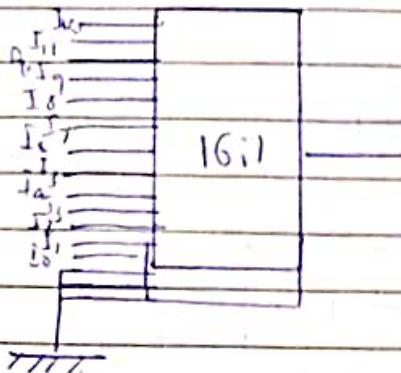
6 : 64 Decoder { Done } core }

HW

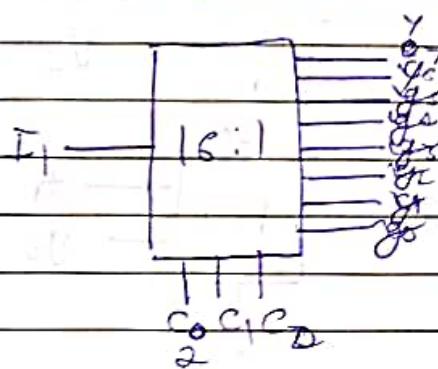
16 : 1 Mux using 2 : 1 only



13:1 MUX



1:8 Demux

C<sub>0</sub> C<sub>1</sub> C<sub>2</sub> C<sub>3</sub>

0 0 0 0

0 0 0 1

0 1 0 0

0 1 0 1

1 0 0 0

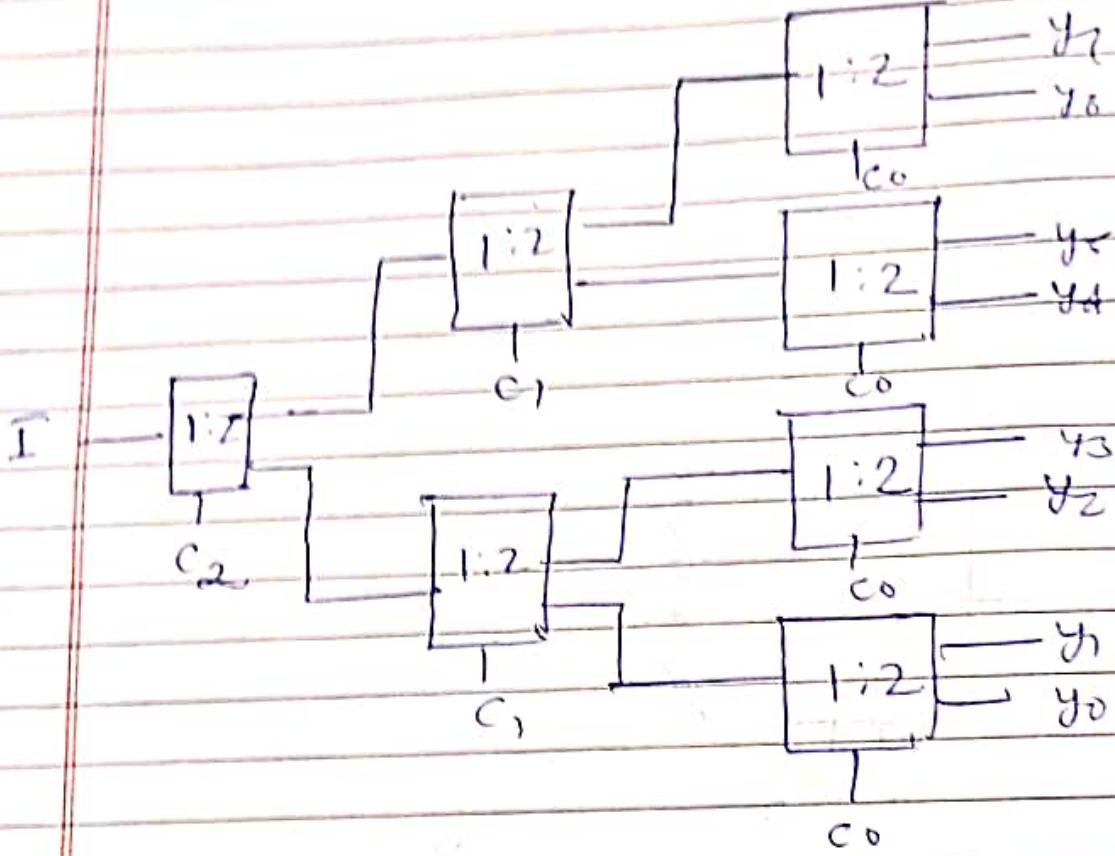
1 0 1 1

1 1 0 0

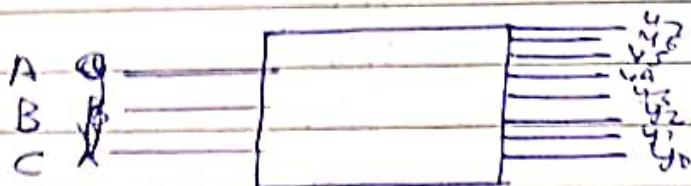
1 1 1 1

Y<sub>0</sub> Y<sub>1</sub> Y<sub>2</sub> Y<sub>3</sub> Y<sub>4</sub> Y<sub>5</sub> Y<sub>6</sub> Y<sub>7</sub>Y<sub>0</sub> Y<sub>1</sub>Y<sub>2</sub>Y<sub>3</sub>Y<sub>4</sub>Y<sub>5</sub>Y<sub>6</sub>Y<sub>7</sub>

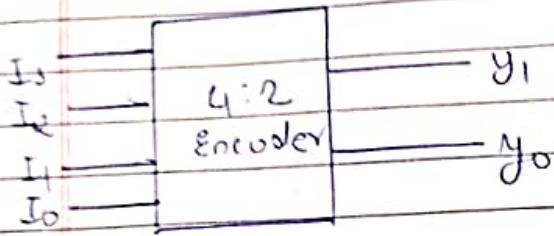
Vis Demo w using Vis Demo only



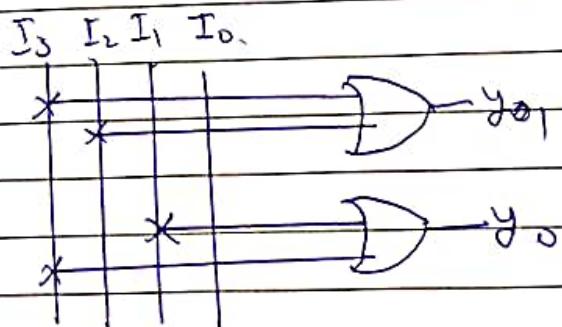
## 3 : 8 Decoder



### En-Coder

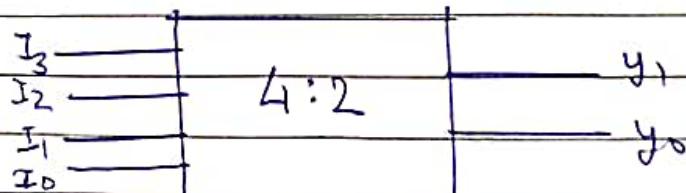


I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>	Y <sub>1</sub>	Y <sub>0</sub>
.	.	1	1	0	0
.	1	.	1	0	1
1	.	1	.	1	0
1	1	.	1	1	1



HW : 8:3 Encoder

### Priority based 4:2 Encoder



$I_3$	$I_2$	$I_1$	$I_0$	$y_1$	$y_0$	
0	0	0	0	X	X	
0	0	0	1	0	0	
0	0	1	0	0	0	
0	0	1	1	0	1	
0	1	0	0	1	0	
0	1	0	1	1	0	
0	1	1	0	1	0	
0	1	1	1	1	0	
1	0	0	0	1	1	
1	0	0	1	1	1	
1	0	1	0	1	1	
1	0	1	1	1	0	
1	1	0	0	1	1	
1	1	0	1	1	1	
1	1	1	0	1	1	
1	1	1	1	1	1	

$I_3$	$I_2$	$I_1$	$I_0$	$y_1$	$y_0$	
00	X	1	0	1	0	
01	1	1	1	1	0	
11	1	1	1	1	1	
10	1	0	1	1	1	

$I_3$	$I_2$	$I_1$	$I_0$	$y_1$	$y_0$	
00	X	1	0	1	0	
01	1	1	1	1	0	
11	1	1	1	1	1	
10	1	0	1	1	1	

19 Apr

BCD to Gray code

Gray code : Convert BCD to Gray  
Keep MSB same

Add 1'st bit of each other

generate any carry over.

Binary

Excess-3 Add 0011 to get 0000

to binary  
add 1'st most point

Complements  
 $(n-1)$ 's complement =  $1, 3, 2, 3, 4, \dots$  complement  
 $n=$  no of digit

Not N = no less want to find complements of  
powers

B Complement addition  
+ve +ve = normal binary addition

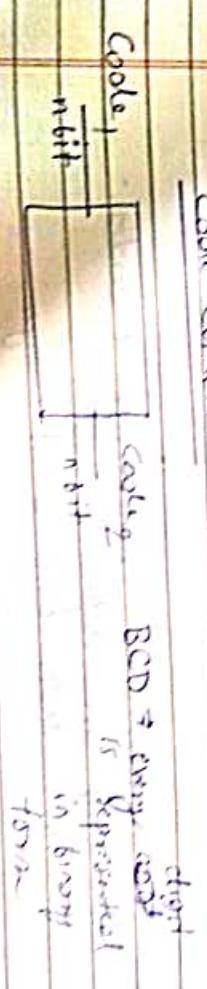
+ve +ve = binary addition  
if 2 is max to 2's  
complement.

To convert +ve -ve use 2's complement  
short

If sum > 9 add 010 to it  
by taking carry to next 4 bit digit

	000	001	010	011	100	101	110	111	101	100
00	0	1	2	3	4	5	6	7	8	9
01	1	2	3	4	5	6	7	8	9	10
10	2	3	4	5	6	7	8	9	10	11
11	3	4	5	6	7	8	9	10	11	12
101	X	1	2	3	4	5	6	7	8	9
100	X	2	3	4	5	6	7	8	9	10
011	0	1	2	3	4	5	6	7	8	9
010	1	2	3	4	5	6	7	8	9	10
001	2	3	4	5	6	7	8	9	10	11
000	3	4	5	6	7	8	9	10	11	12

Code Converter



BCD : 4 bit representation  
if  $(num)_{10} < 9$  binary  
if  $(num)_{10} > 9$  each digit binary + 1 bit

Addition in BCD : Normal binary addition

If sum > 9 add 010 to it  
by taking carry to next 4 bit digit

Binary

<u>Dec No.</u>	<u>Binary</u>	<u>BCD</u>
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	00010000 (1010)BCD
11	1011	00010001 (1011)BCD
12	1100	00010010 (1100)BCD
13	1101	00010011 (1101)BCD
14	1110	00010100
15	1111	00010101

$$(10)_2 = (010)_6 = (00010000)_{BCD}$$

$$(13)_2 = (110)_6 = (00010011)_{BCD}$$

Cyclic codes

- + Cyclic codes are non-weighted codes.
- + They are effective codes.
- + Take only 1 bit change.

Answers  
Solve

Q2 Judge the code converter which takes binary as inputs and produces gray code as output. (2.5t)

Hyp 4/3/18

Q

Q2 Convert  $(1001)_2$  to gray code

$\begin{array}{r} 1 \ 0 \ 0 \ 1 \\ - \\ \end{array}$   
↓  
 $\begin{array}{r} 1 \ 0 \ 0 \ 1 \\ - \\ \end{array}$

$(1101)_2$  gray code

Q2 Convert gray code with binary  $(1101)_2$  gray code. G = EPL

$\begin{array}{r} 1 \ 1 \ 0 \ 1 \\ | \ 0 \ 1 \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \ 1 \end{array}$   
↓  
 $\begin{array}{r} 1 \ 1 \ 0 \ 0 \ 1 \\ | \ 0 \ 1 \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \ 1 \end{array}$   
↓  
 $\begin{array}{r} 1 \ 1 \ 0 \ 0 \ 1 \\ | \ 0 \ 1 \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \ 1 \end{array}$

I/P			O/P		
A	B	C	$\frac{A}{2}$	$\frac{B}{2}$	$\frac{C}{2}$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Wrap  $\rightarrow$   $B_1, B_2$

Excess - 2

$\rightarrow$  Binary ~~(3)~~ D

Pad the code consider that there is  
2 greatest 4 bit excess 3

I/P				O/P			
R	B	C	D	$B_2$	$B_1$	$B_0$	$B_3$
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	1
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	1
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	0

1 1 0 1

1 1 1 0

1 1 1 1

(INCURRED)

~~X A B C D E P~~

X

X

## Q) Quine-McCluskey Method

e.g., Obtain the minimal expression for  
 $f = \sum m(0, 1, 6, 7, 8, 9, 13, 14, 15)$

Step 1 : obtain index of and arrange the minterms accordingly.

Minterms	Binary Equivalent	Index
0	0000	0
1	0001	1
6	0110	2
7	0111	3
8	1000	1
9	01001	2
13	01101	3
14	1110	3
15	1111	4

Step 2:

Index	minterms	Binary	Points	A'BC'D'	A'B'C'D'
0	0	0000	(0,1)	0001	(0,1,3)
1	1	0001	(0,8)	X 00D	(0,1,3,5)
	8	1000	(1,9)	X 001	(1,4,5)
2	6	0110	(5,9)	0100X	
	9	1001	(C,7)	011X	
3	7	0111	(G,14)	X 110	1X01
	13	1101	(9,13)	X 111	
4	14	1110	(13,15)	(18,15)	11X1
	15	1111	(14,15)	(14,15)	111X

$$\text{Final Output} = \overline{B}\overline{C} + \overline{B}C$$

Step 3:

Final output is  $((0,1,8,9), 0(6,7,14,15), (9,13) \& (13,15))$

~~Step 3:~~

<del>X 00X</del>	0	1	8	9	$\rightarrow X 00X \rightarrow \overline{B}C$
<del>X 00X</del>	6	7	14	15	$\rightarrow X 11X \rightarrow BC$
<del>X 00X</del>	9	13			$\rightarrow 1X01 \rightarrow A'CD$
<del>X 00X</del>	13	15			$\rightarrow 110X1 \rightarrow ABD$

SOP:  $\overline{B}\overline{C} + BC + A'CD + ABD$

Step 4:

Youtube

Half-Mux with Don't care

$$F(A, B, C, D) = \sum m(0, 2, 6, 11, 12, 13) + \sum d(3, 8, 14)$$

A	B	C	D	Y	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	I <sub>7</sub>	
0	0	0	0	1	D	0	2	4	6	8	10	12	14
0	0	0	1	0	D	1	3	5	7	9	11	13	15
0	0	1	0	1				D	1	0	D	0	D
0	0	1	1	X									
0	1	0	0	0									
0	1	0	1	0									
0	1	1	0	1									
1	0	0	0	X									
1	0	0	1	0									17
1	0	1	0	0									
1	0	1	1	1									
1	1	0	0	1									
1	1	1	0	X									
1	1	1	1	0									

For POS:

$$A \quad 1 \rightarrow \bar{A}$$

$$0 \rightarrow A$$

## Flip Flops

- Binary cell used to store 1 bit of data.
- Sequential ckt  
(output = present input + ~~last~~ past output)
- Contains memory element to store past output
- Has 2 outputs
  - Normal &
  - complement

### Type

SR / Basic flip flops

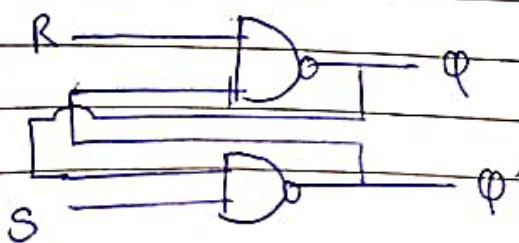
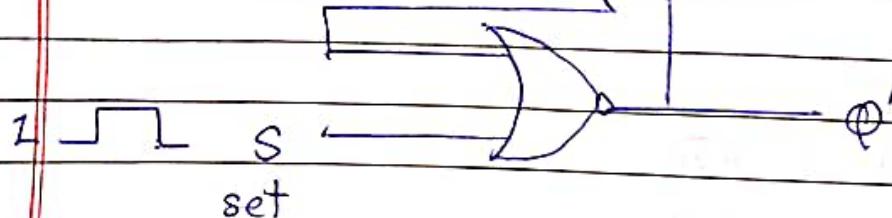
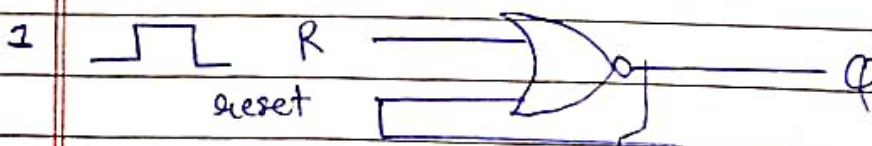
J-K Flip Flops

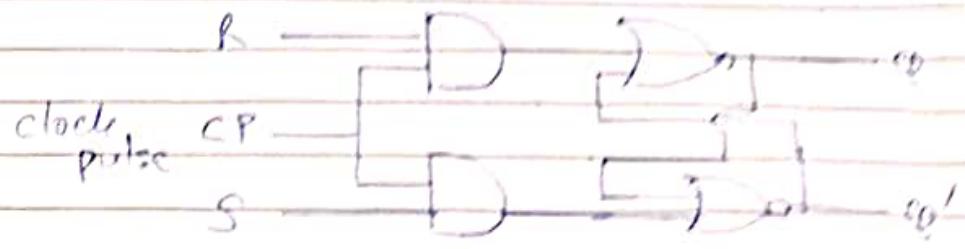
Delay Flip Flops

Toggle Flip Flops

### SR Flip Flop

SET - RESET flip flop is formed by using  
2 NOR / 2 NAND gates





output  
output

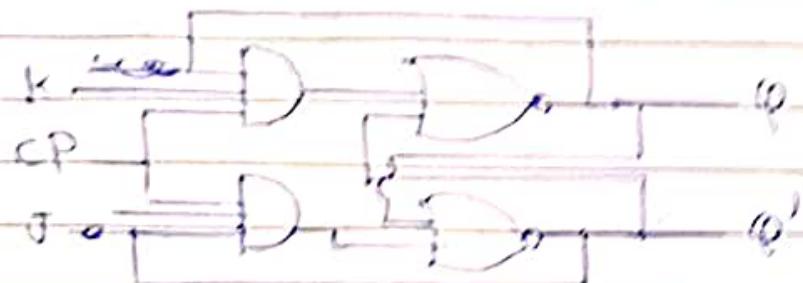
S	R	$Q(N)$	$Q(N+1)$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	X	X

\* present output

$Q(N+1) \rightarrow$  previous output

S	R	$Q(N)$	$Q(N+1)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

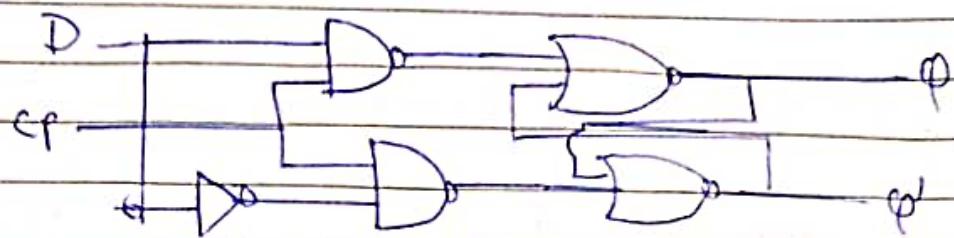
Characteristic eqn :  $Q(N+1) = S + \bar{R} Q(N)$   
 ( Using K map 3 variables )

D-K Flip Flops (Refinement of SR Flip Flop)Truth Table

J	K	$Q(N+1)$	
0	0	$Q(N)$	Held
0	1	0	Clear to 0
1	0	1	Clear to 1
1	1	$Q'(N)$	Toggle

Characteristic eqn:  $J\bar{Q}'(N) + \bar{K}Q(N) = Q(N+1)$ Delay flip flop

- \* Transparent latch flip flops
- \* I/P & O/P are equal
- \* SR flip flops are converted into D flip flops by applying inverter

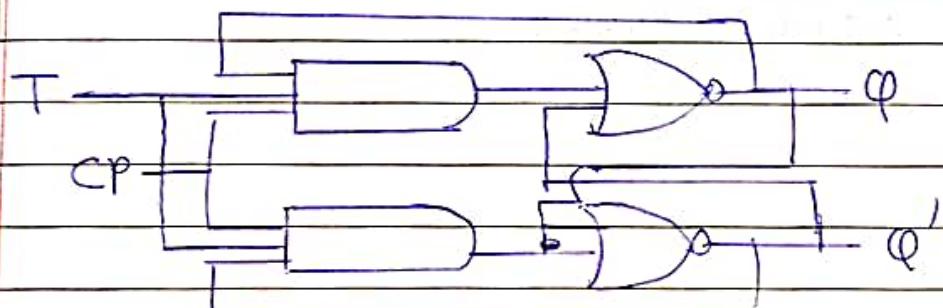


D	$Q(N+1)$
0	0
1	1

Clear to 0  
Clear to 1

Characteristic eqn  
 $Q(N+1) = D$

### Toggle Flip Flop



T	$Q(N+1)$
0	$Q(N)$
1	$\bar{Q}(N)$

$$Q(N+1) = \bar{T}Q(N) + T\bar{Q}(N)$$

### Flip Flop Vs Latch

\* Always has a clock signal

Doesn't have a clock signal

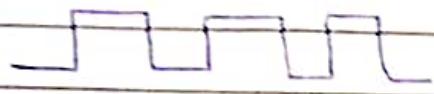
\* Checks input but changes output only defined by clock signal

Changes output immediately after input change

Classified as synchronous /  
asynchronous

Edge triggered  
device

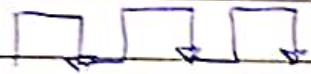
Level triggered  
device



level trigger



Positive trigger



Negative trigger

Step 3

The final outcome of Quine-Mccluskey table  
 $(0, 1, 8, 9)$

Step 4

Prime Equivalent chart

Product terms

Patterns minterms

$\textcircled{0}$	$\textcircled{1}$	$\textcircled{6}$	$\textcircled{7}$	$\textcircled{8}$	$\textcircled{9}$	$13$	$14$	$15$
-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	------	------	------

select  $0 \ 1 \ 8 \ 9$ 

$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
--------------	--------------	--------------	--------------	--------------

all terms are covered  
 select  $6 \ 7 \ 14 \ 15$

select  $6 \ 7 \ 14 \ 15$ 

$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
--------------	--------------	--------------	--------------

 $\checkmark \ 9 \ 13$ 

$\checkmark$	$\checkmark$
--------------	--------------

 $13 \ 15$ 

$\checkmark$	$\checkmark$	$\checkmark$
--------------	--------------	--------------

Prime implicants:  $\{0, 1, 6, 7, 8, 14\}$   
 $n = 6$

$\rightarrow (0, 1, 8, 9), (6, 7, 14, 15), (9, 13)$

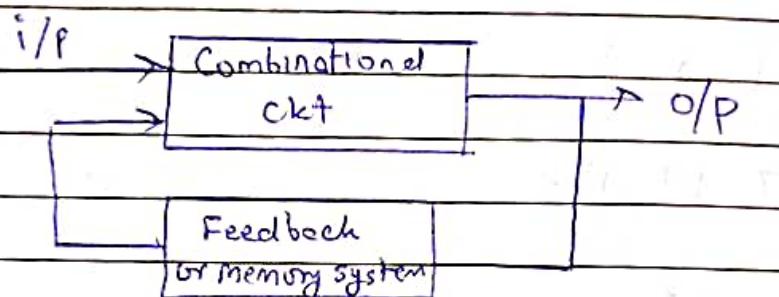
HW: Solve 4 variable problems of kmap  
 with Quine-Mccluskey method

## Sequential Circuit Design

+ Combinational ckt's are the ckt's whose output at any given time depends upon the present inputs only.

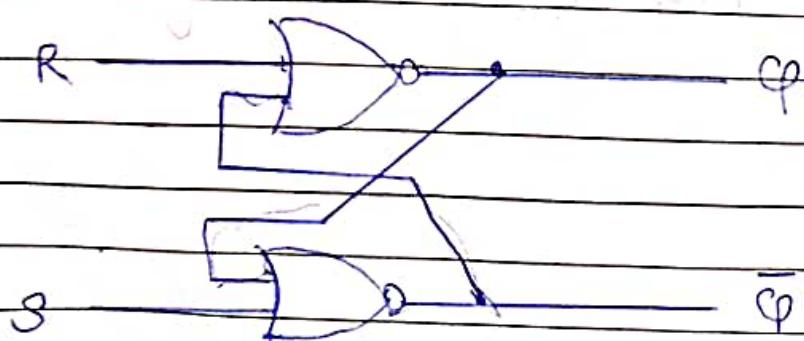
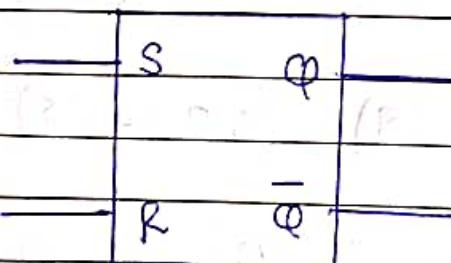
In a sequential ckt the output depends upon the present input as well as the past output

e.g. →



## Latches & Flip Flops

### S-R Latch

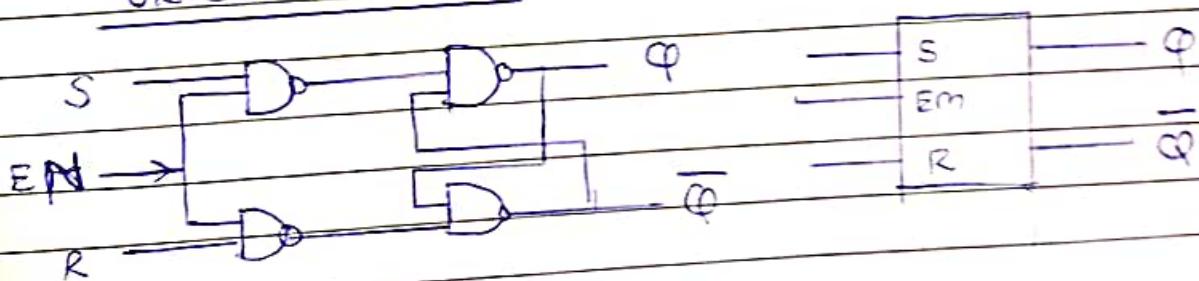


Excitation : The input signal combinations which commands the flip-flop to change the state are called as excitations.

Input		Present O/p ( $Q_n$ )	Next O/p ( $Q_{n+1}$ )	Output
S	R	$Q_n$	$Q_{n+1}$	
0	0	0	0	No change
0	0	1	1	
0	1	0	0	Reset -
0	1	1	0	
1	0	0	1	Set
1	0	1	1	
1	1	0		Invalid X
1	1	1		Invalid X ]

HW : Design SR latch with NAND gates.

Created SR latch



EN	S	R	Q	-Q
1	0	0	Same as above Latch for EN = 1	
1	0	1	EN = 0 No change	

HW  $\Sigma m(0, 1, 2, 3, 4, 7, 9, 13)$

Index

<u>Step 1</u>	0	0000	0	0, 1, 2, 3, 4, 7, 9, 13
	1	0001	1	
	2	0010	1	0, 1, 2, 3, 4, 7, 9, 13
	3	0011	2	0, 4, 7, 9, 13
	4	0100	1	1, 9, 13
	7	0111	3	3, 7, 13
	9	1001	2	9, 13
	13	1101	3	

Step 2

(0, 1, 2, 3)

~~(0, 2)~~

(0, 4)

(3, 7) ~~(1, 3)~~ (9, 13)

Min Term	Index	Bin eq.	Pair	Pair	Pair
0	0	0000	(0, 1)	000X (0, 1)	000X
1	1	0001	(0, 2)	00X0 (0, 2)	00X0
2	1	0010	(0, 4)	0X00 (0, 4)	0X00
3	2	0011	(1, 4)	X000 (1, 3)	00X1
4	1	0100	(1, 2)	00X1 (1, 9)	X001
7	3	0111	(1, 3)	00X1 (2, 3)	00X1
9	2	1001	(2, 3)	001X (3, 7)	0X11
13	3	1101	(2, 4)	001X (9, 13)	1X01
			(3, 7)	(3, 7)	
			(9, 13)	(9, 13)	

$$\cdot \bar{A}B \quad \bar{A}\bar{C}D \quad (\cancel{0, 1, 2, 3}) \quad (0, 1, 2, 3)$$

$$00XX \quad 0X00$$

$$(0, 1, 2, 3), (\cancel{0, 1, 2}), (0, 4) \cancel{(1, 3)} \quad (0, 2, 1, 3)$$

$$(1, 9) \quad (3, 7) \quad (9, 13) \quad (\cancel{3, 7, 9, 13})$$

$$X001 \quad 0X11 \quad 1X01$$

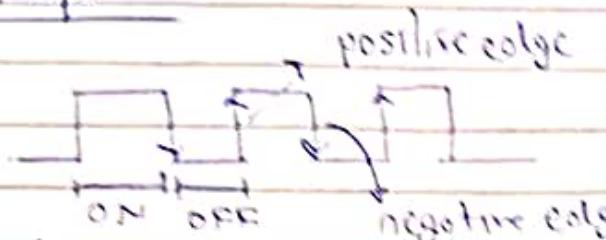
$$\bar{B}CD \quad \bar{A}CD \quad \bar{A}\bar{C}D$$

Min	B.	Index	Point		
-0	0000	0	(0,1)	000X	(0,1,8,15)
1	0001	1	(0,8)	X00'D	(0,8,15)
3	0011	2	(1,3)	00X1	(1,3,5,7)
5	0101	2	(1,5)	0X01	(1,3,9,11)
7	0111	3	(1,9)	X001	(1,5,3,7)
8	1000	1	(3,7)	0X11	(1,9,3,11)
9	1001	2	(3,11)	X011	(3,7,9,11)
11	1011	3	(5,7)	01*x1	(5,7,3,11)
13	10101	3	(8,9)	100X	x1x1
15	1111	4	(9,13)	10X1	(9,11,3,15)
			(11,15)	1X11	1XX1
			(13,15)	11X1	(9,11,13,15)
					1XX1

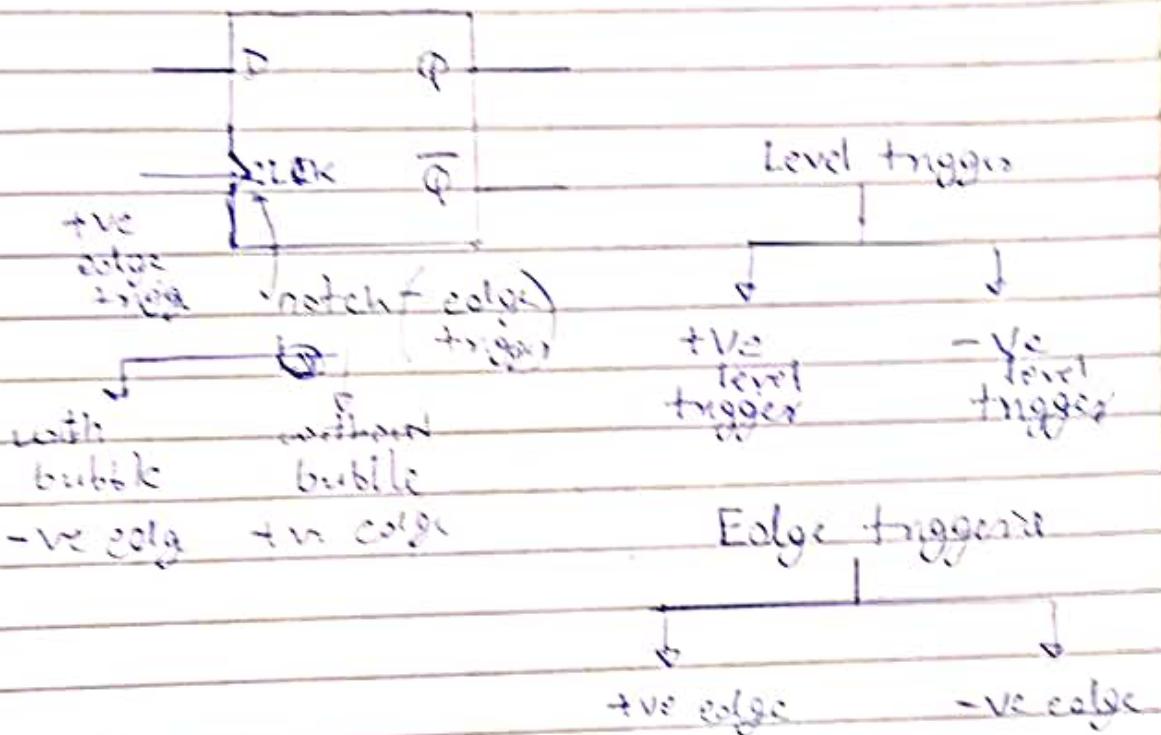
0 6 10 8 9	X 00 X	(0,1,8,9,11,3,5,7)
0 6 8	0	⊗
1 3 5 7	X 00 1	
1 3 9 11	X 0 X 1	(1 3 9 11)
3 7 11 15	X X 1 1	5,7,13,15
5 7 13 15	X 1 X 1	X X X 1
9 11 13 15	1 X X 1	
4 3 5 7 9 11 13 15	D	0,1,3,5,7,9,11,15
1 3 9 11		vvvvvvvv
3 7 11 15		vv vv
5 7 13 15		v v v v
0 1 8 9		vv vv

BC + D

## Flip-Flops



$$\left\{ \text{duty cycle} = \frac{\text{ON time}}{\text{OFF time}} \times 100\% \right\}$$



edge	Clock	D	Input	Output	State
-	-	X		Q → 0	No change
↑	↑	X		1 → 1	No change
↓	↓	0		0 → 0	Reset state
↓	↓	0		1 → 0	
↑↓	↑↓	0		0 → 1	Set state
↑↓	↑↓	1		1 → 1	

Registers  
Shift Registers

Types:

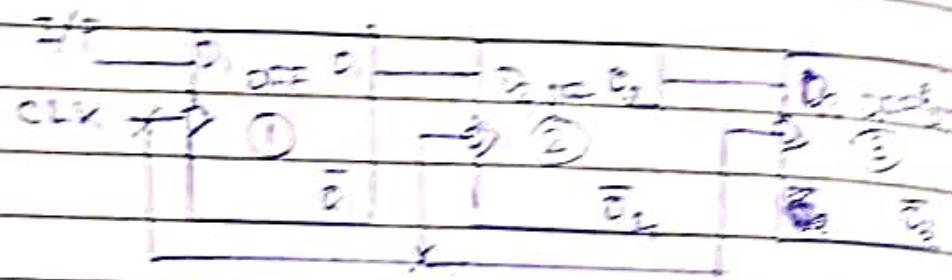
1) SR5D

 $S + S_{not}$  $S + S_{not}$  $S + S_{not}$ 

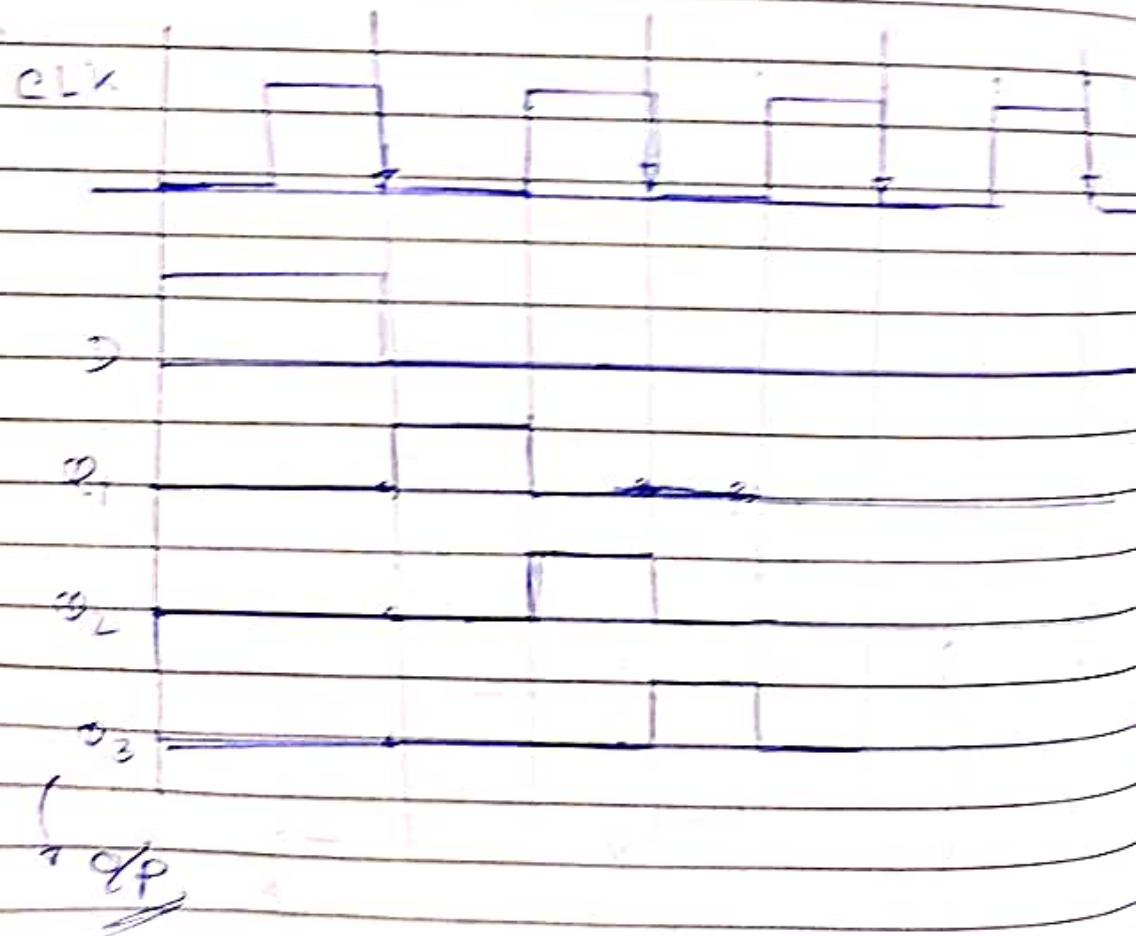
2) SRPS

3) PR5D

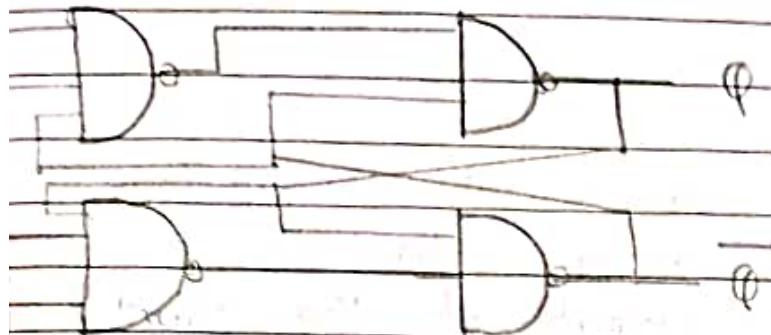
4) PIPD



Initially  $Q_1 = Q_2 = Q_3 = 0$



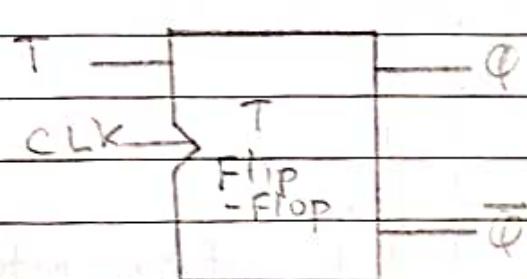
## J-K Flip-Flop



Positive edge

J	Q	J-K	CLK	J	K	$Q_n$	$Q_{n+1}$
—	—	Flip Flop	↓	x	x	$Q_n$	$Q_n$
—	—	—	↑	0	0	$Q_n$	$Q_n$ <small>Memory</small>
—	—	—	↑	0	1	0	0 <small>Reset</small>
—	—	—	↑	1	0	1	1 <small>Set</small>
—	—	—	↑	1	1	0	1 <small>Toggle</small>

To get T-Flip Flop  
we short J and K  
replace J/K with T



\* Truth Table will only have set <sup>memory</sup> and toggle states.  
{No set / Reset states}

CLK	T	$Q_n$	$Q_{n+1}$
↓	0 X		
↑	1 0	$Q_n$	$Q_n$ (Memory)
↑	1	0 → 1	0 (Toggle)

## Use of Flip Flops

### Counters

A digital counter is a set of flip flops whose state changes in response to the clock pulse applied at the input of the counter.

The flip flops are interconnected such that the combined state at any time is the binary equivalent of total <sup>number of</sup> pulses that have occurred upto that time.

Counters can be -

- i) Asynchronous (Ripple)
- ii) Synchronous

Modulus: The number of states through which the counter passes before returning to the starting state is called as modulus of the counter.

2 flip-flops

Example: A 2 bit counter can count

[ 0 0

0 1

1 0 ]

1 1 ]

before returning to the initial state (0,0). Then modulus of the counter is 4.

A full modulus counter is the counter which counts all the possible states before restarting. While on the contrary if any counter counts lesser number of states than the full modulus then such counters are called as variable modulus counter.

Terminal Count: The final state of the counter sequence is called as Terminal count.

Initial State: The state from where the counting starts, as well as the counter returns after counting is called as initial state.

Lock-out: For variable modulus counter the problem of lock-out may occur where counter will get stuck up in the invalid states. When it gets stuck in the invalid state, the counter becomes useless. Lock-out prevention is necessary for variable modulus counter.

Examples:

- ① Person counting in a mall
- ② Counting number of bottles in a bottling plant
- ③ Counting number of pulses in car's braking system.

## Asynchronous / Ripple Counter

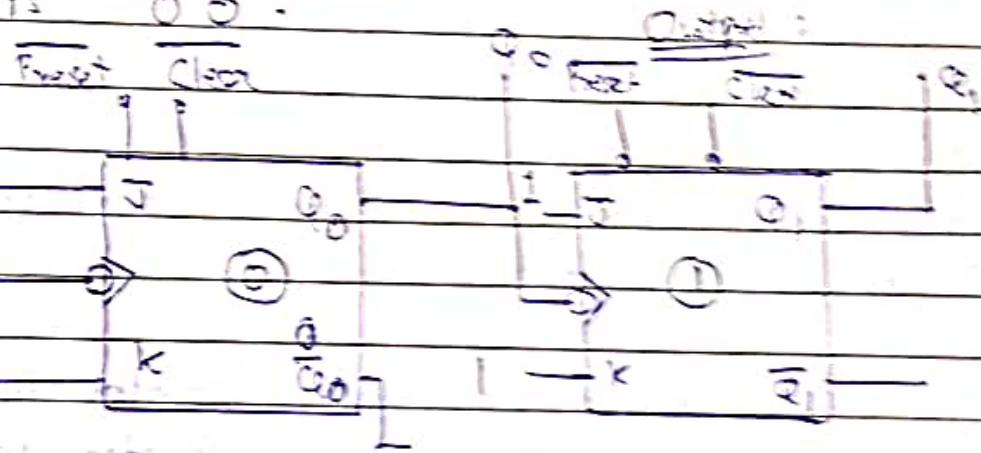
Design a 2-bit asynchronous counter.

Truth table:

- + Nothing is specified here
- Up counter
- Full modulus

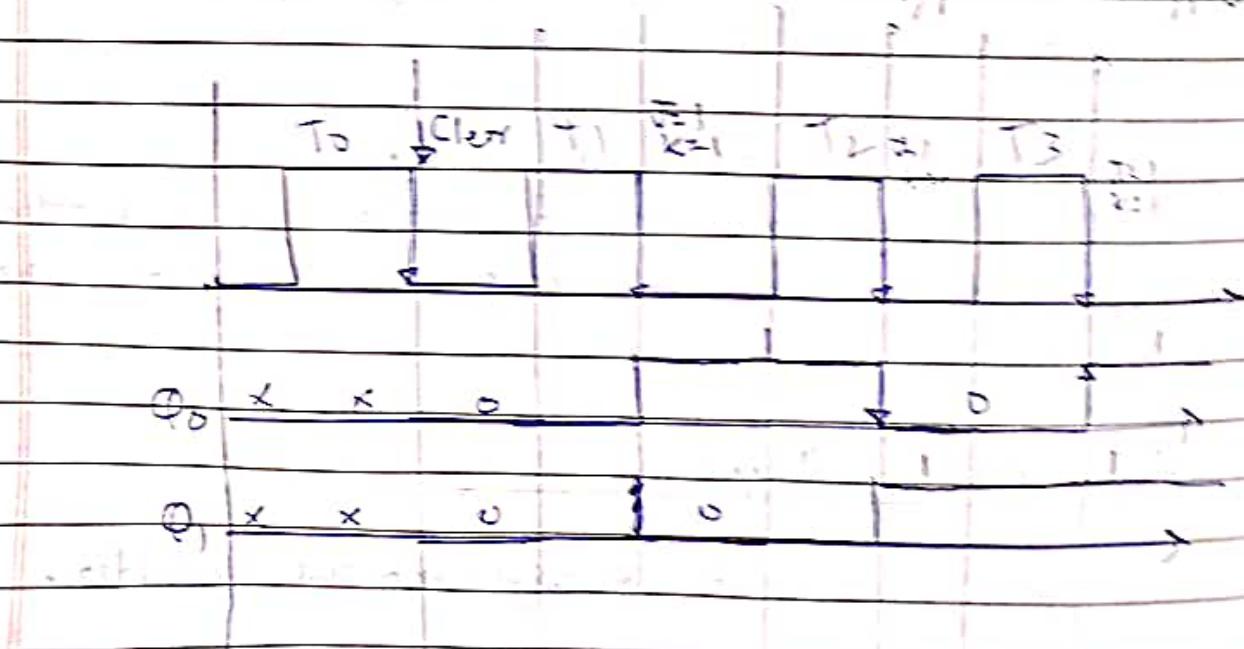
OR,

Design = 2-bit asynchronous counter where initial value is 00.



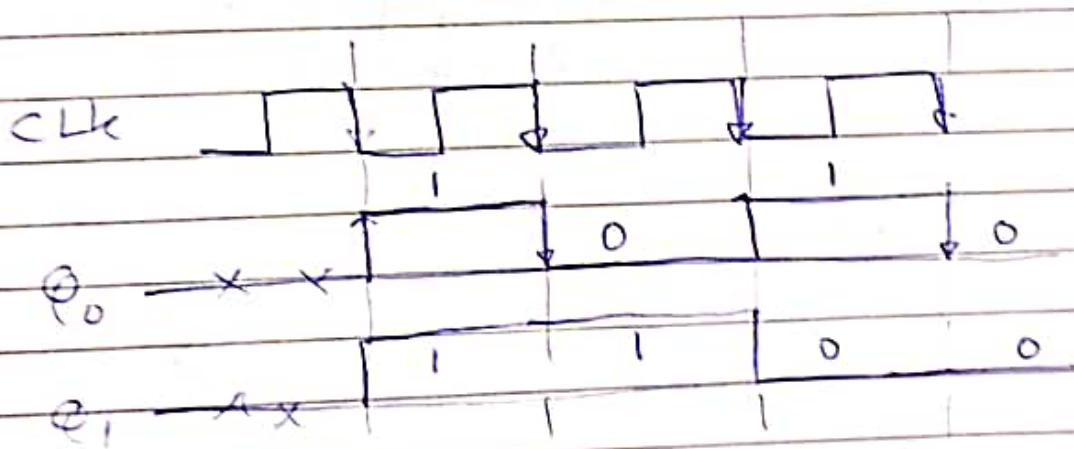
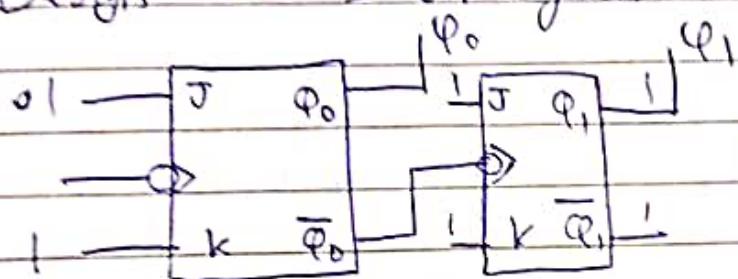
Inputs: + Use OV / ground to activate preset and clear values zero or one.

Outputs: Preset S/P & Clear C/P



CLK	$Q_1, Q_0$
T <sub>01</sub>	0 0
T <sub>12</sub>	0 1
T <sub>23</sub>	1 0
T <sub>34</sub>	1 1
repeat $\rightarrow$ T <sub>45</sub>	0 0

Q) Design a 2-bit asynchronous down counter

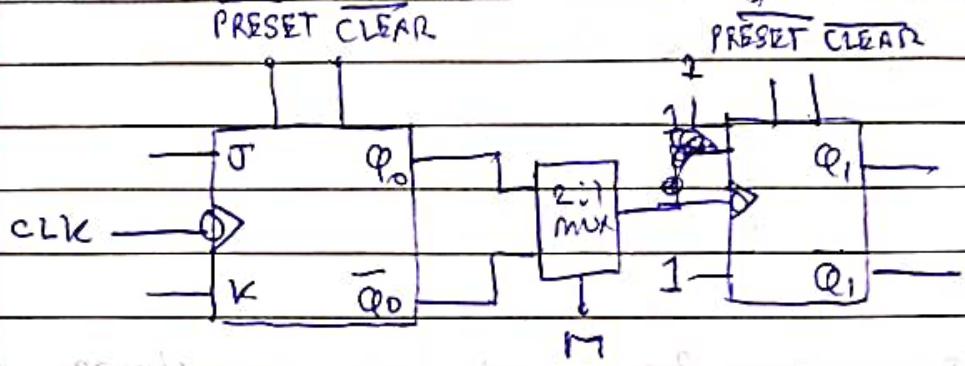


T	$Q_1, Q_0$
T <sub>1</sub>	1 1
T <sub>2</sub>	1 0
T <sub>3</sub>	0 1
T <sub>4</sub>	0 0

Q) Design 2 bit up-down counter with max control bit.

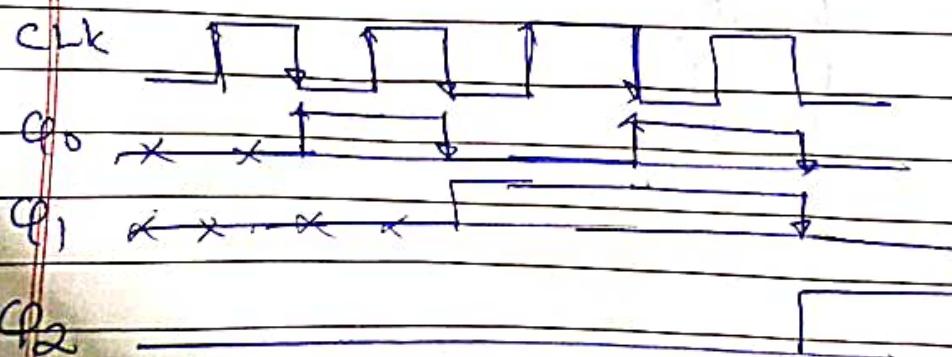
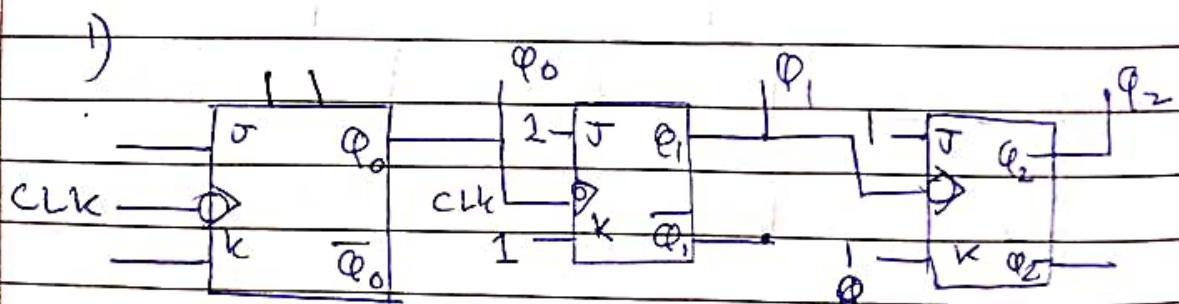
Steps:

Assuming  $M=1$  then up counter  
 $M=0$  then down counter

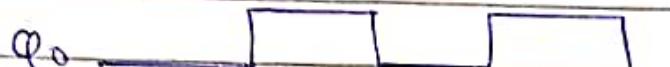
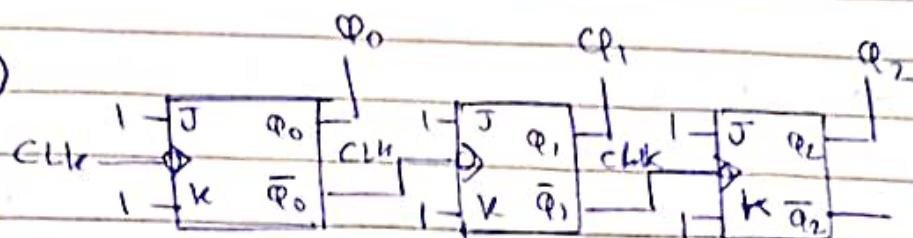


- HW:
- 1) Design 3 bit up counter
  - 2) Design 3 bit down counter
  - 3) Design 3 bit up-down counter

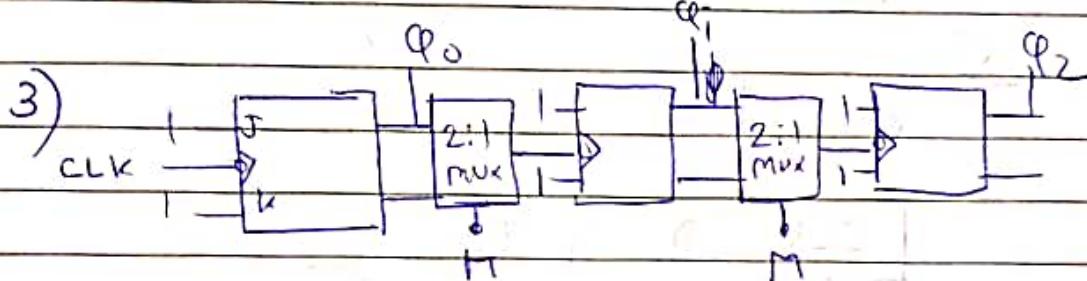
HW: 1) Design 4 bit up counter, down counter & up-down counter.



2)



3)



## Asynchronous Variable Modulus Counter

Design MOD = 6 counter

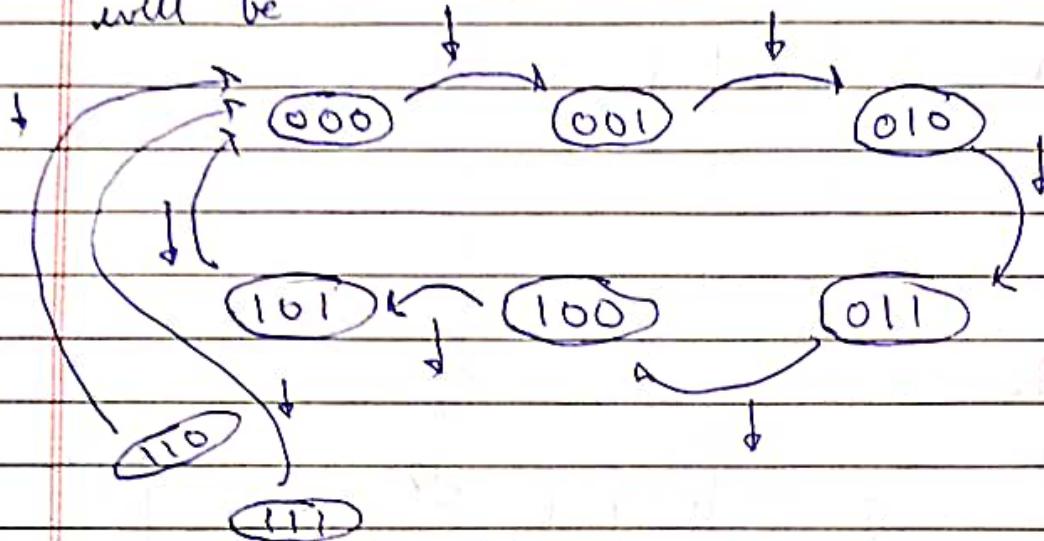
Since it is a mod 6 counter it requires 3 bits. We use -ve edge triggered T flip-flops. Hence 3 flip-flops.

Initial state = 000

Since it is MOD 6 with initial state as 000 we have final state as 101. The initial state is 000 and when clock arrives it goes to 001.

This process finishes if 101. After that counter resets to initial state.

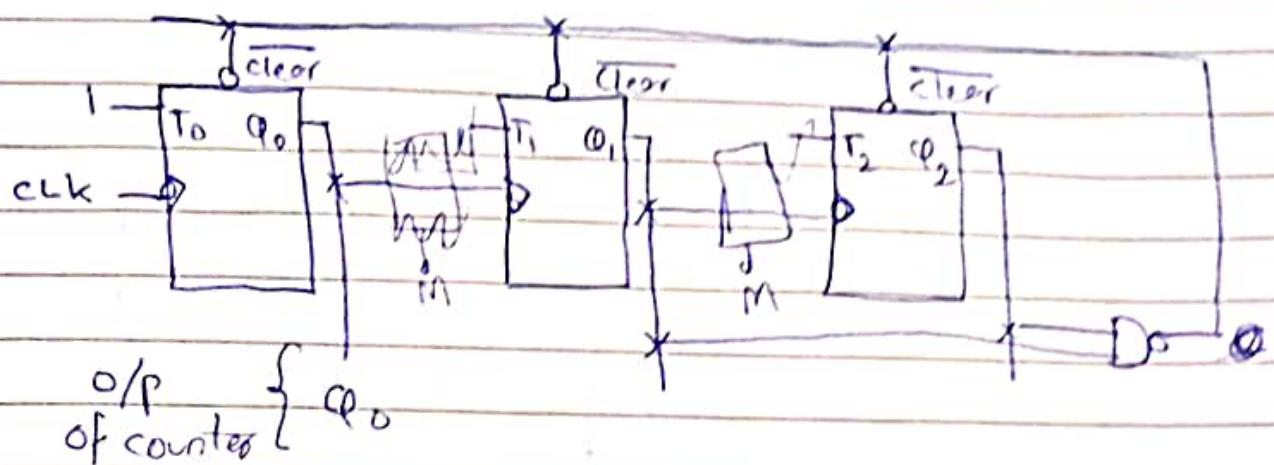
Therefore states 000 to 101 are valid states.  
while 110 and 111 are invalid states. So  
the state diagram for this MOD 6 counter  
will be



CLK	$Q_2$	$Q_1$	$Q_0$	C	M
↓	0	0	0	0	0
↓	0	0	1	0	0
↓	0	1	0	0	0
↓	0	1	1	0	0
↓	1	0	0	0	B
↓	1	0	1	0	B
	1	1	0	1	0
	1	1	1	1	0

$Q_2$	$Q_1$	$Q_0$	11	10
$Q_0$	0	1	2	3
0	0	1	2	3
1	1	1	3	17

$$C = Q_2 Q_1$$



HW: MOD 6 down counter  
MOD 6 updown counter

M  $Q_2$   $Q_1$   $Q_0$  C

1 0 0 0 0

1 0 0 1 0

1 0 1 0 1

1 0 1 1 1

1 1 0 0 1

1 1 0 1 1

1 1 1 1 1

up counter

Mod 6

$Q_1 Q_0$	00	01	11	10
01	0	4	14	13
11	1	5	15	19
10	3	1	12	10
11	2	0	13	11

Design

HW: ~~Design~~ MOD 10 counter

(BCD counter)

(Divide by 10)

counter

(Mod number)

$$C = \bar{T}_1 Q_2 \bar{Q}_1 + T_1 \bar{Q}_2 \bar{Q}_1$$

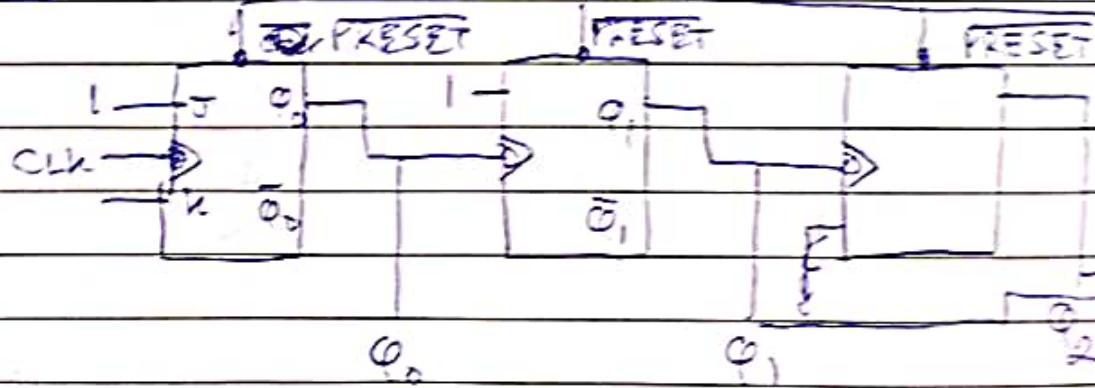
Design mod 8 up counter

HV

$Q_3\ Q_2\ Q_1\ Q_0$	clk	$\bar{Q}_3\ \bar{Q}_2\ \bar{Q}_1\ \bar{Q}_0$	C
0 0 0 0	+	0 0 0 0	1
0 0 0 1	+	0 0 1 0	1
0 0 1 0	+	0 1 0 0	0
0 0 1 1	+	0 1 1 0	0
0 1 0 0	+	1 0 0 0	0
0 1 0 1	+	1 0 1 0	0
0 1 1 0	+	1 1 0 0	0
0 1 1 1	+	1 1 1 0	0

~~$Q_3\ Q_2\ Q_1\ Q_0$~~   
~~0 0 0 0~~  
~~0 0 0 1~~  
~~0 0 1 0~~  
~~0 0 1 1~~  
~~0 1 0 0~~  
~~0 1 0 1~~  
~~0 1 1 0~~  
~~0 1 1 1~~

Q:  $C = \bar{Q}_2\bar{Q}_1$



$Q_3\ Q_2\ Q_1\ Q_0$	C
0 0 0 0	0
0 0 0 1	0
0 0 1 0	0
0 0 1 1	0
0 1 0 0	0
0 1 0 1	0
0 1 1 0	0
0 1 1 1	0

QUESTION NO.  
1

$S_3 S_2 S_1 S_0$       C

1 0 0 0      0

1 0 0 1      0

1 0 1 0      10

1 0 1 1      11

1 1 0 0      10

1 1 0 1      11

1 1 1 0      10

1 1 1 1      11

~~0000 0000~~

~~0000 0001~~

~~0000 0010~~

~~0000 0011~~

~~0000 0100~~

~~0000 0101~~

~~0000 0110~~

~~0000 0111~~

~~0000 1000~~

~~0000 1001~~

~~0000 1010~~

~~0000 1011~~

~~0000 1100~~

~~0000 1101~~

~~0000 1110~~

~~0000 1111~~

~~0001 0000~~

~~0001 0001~~

~~0001 0010~~

~~0001 0011~~

~~0001 0100~~

~~0001 0101~~

~~0001 0110~~

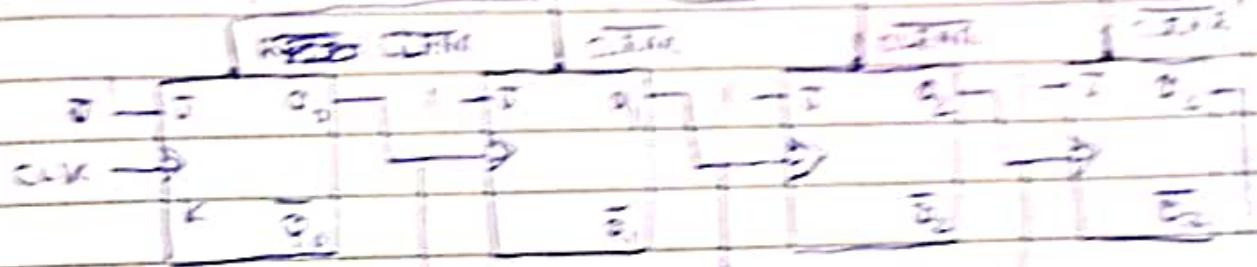
~~0001 0111~~

~~0001 1000~~

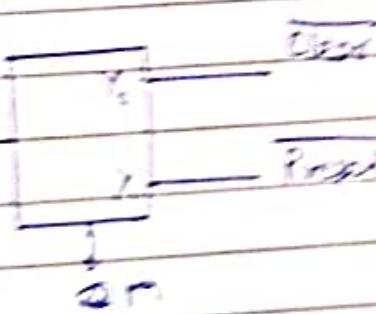
~~0001 1001~~

~~0001 1010~~

~~0001 1011~~



BCD  
Counter



4 bit BCD Counter

4 bit BCD Counter

Excitation Table~~S = R~~SR Flip Flop

$Q_n$	$Q_{n+1}$	S	R		S	R
0	0	0	0 X	$\Rightarrow$	0 X	
0	1	1	0	$\Rightarrow$	1 0	
1	0	0	1	$\Rightarrow$	0 1	
1	1	0	X 0	$\Rightarrow$	X 0	
			1			

~~Qn~~J-K Flip-Flop

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

Toggles

$Q_n$	$Q_{n+1}$	T
0	0	0 0
0	1	1
1	0	1
1	1	0

XOR gate

## Synchronous Counter

Asynchronous counters are serial counters and hence they are slow in operation. In these counters delays are accumulated at every stage of flip flop. Many a times it is observed that when an asynchronous counter is used for high speed operations then it may skip a few states which lead to malfunction.

Therefore to solve this problem synchronous counters are used. However designing of synchronous counters is a complex task.

Design a 3 bit synchronous counter where pulses are given simultaneously.

Step 1 :-

Determine the number of flip flops required.

Since it is 3 bit counter 3 flip-flops are required.

It is full modulus counter hence 8 states. Initial state

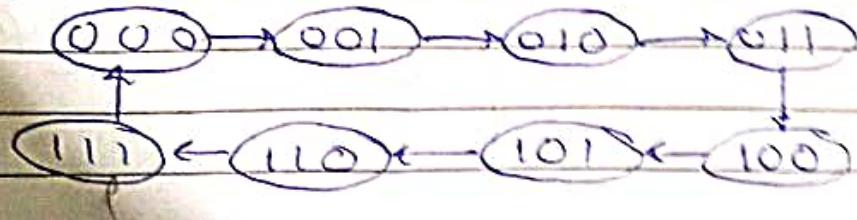
is 0 0 0. Final state is 1 1 1.

It is an up counter.

Step 2:-

Valid States : 000, 001, 010, 011, 100, 101, 110, 111

Invalid State: N/A



Step 3:

Selection of flip flops & its excitation table.

For this designing -ve edge trigger with J-K  
Flip-Flop is being used as follows-

P.S	N.S	Reqd	Set
$Q_n$	$Q_{n+1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Step 4:

Minimal expressions for counter

PS	NS	Req i/p
$Q_2 Q_1 Q_0$	$Q_2 Q_1 Q_0$	$J_2 \quad k_2$
0 0 0	0 0 0	0 X      0 X
0 0 1	0 1 0	0 X      1 X    X 1
0 1 0	0 1 1	0 X      X 0    1 X
0 1 1	1 0 0	1 X      X 1    X 1
1 0 0	1 0 1	X 0      0 X    1 X
1 0 1	1 1 0	X 0      1 X    X 1
1 1 0	1 1 1	X 0      X 0    1 X
1 1 1	0 0 0	X 1      X 1    X 1

$Q_2 Q_1$	00	01	10	11	$Q_2 = Q_1 Q_0$
0	0	2	X <sub>2</sub>	X <sub>1</sub>	J <sub>2</sub>
1	1	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>	J <sub>2</sub> = Q <sub>1</sub> Q <sub>0</sub>
1	1	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>	

$$k_2 = \bar{Q}_1 Q_0$$

$$\textcircled{1} J_1 = Q_0$$

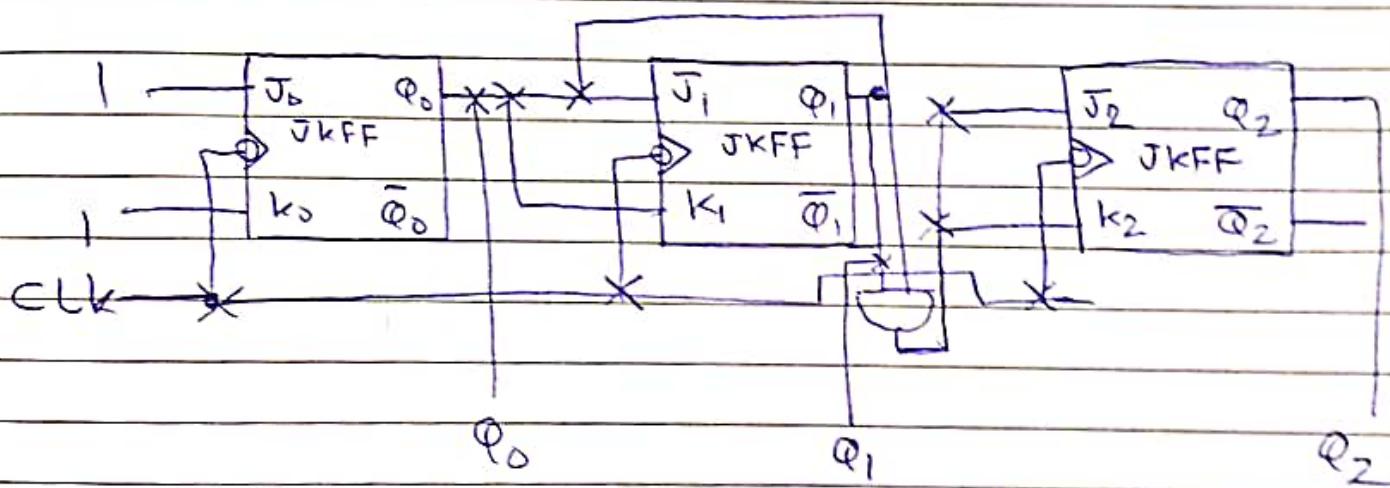
$$\textcircled{2} k_1 = \bar{Q}_0$$

$$J_0 = 1$$

$$k_0 = 1$$

$Q_0$	0	0	0	1	1	1	0
$\bar{Q}_0$	1	1	1	0	X	X	1
$J_1$	1	X	X	1	X	X	1
$\bar{J}_1$	X	1	X	1	1	X	1
$J_0$	1	X	X	1	1	X	1
$\bar{Q}_1$	X	1	X	1	1	X	1
$J_2$	1	X	X	1	1	X	1
$\bar{Q}_2$	X	1	X	1	1	X	1

Step 5: Implement the circuit



HW: Design 3 bit synchronous down counter

Eg) Design MOD = 6 synchronous up counter.

Step 1: 3 flip flop  
 T flip flop is used  
 - Vc edge triggered

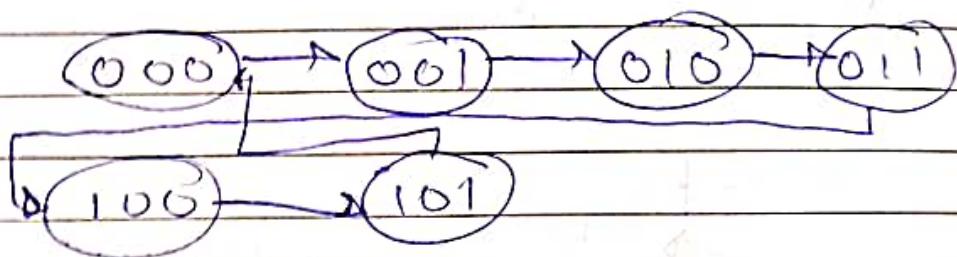
Step 2:

Valid states:

0 0 0	0 0 1	0 1 0	0 1 1	1 0 0
1 0 1	<del>KDD</del>	<del>EDD</del>		

∅ Invalid states:

~~0 0 1 1 0~~    1 1 1



Step 3:

∅ T flip flop

$Q_n$	$Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

Step 4:

$Q_n$	$Q_{n+1}$	$T_2$	$T_1$	$T_0$
0 0 0	0 0 1	0	0	1
0 0 1	0 1 0	0	1	1
0 1 0	0 1 1	0	0	1
0 1 1	1 0 0	1	1	1
1 0 0	1 0 1	0	0	1

$\Phi_0$ 

1	1	0
1	1	1

 $\Phi_{M+1}$ 

0	0	0
0	0	0

 $T_2$ 

1	1	0
1	1	1

 $T_1$  $T_b$  ~~$\Phi_0$~~  $\Phi_2 \oplus \Phi_1$ 

$\Phi_0$	0	1	1	1
1	0	1	1	1
$\Phi_1$	1	0	1	1
$\Phi_2$	1	1	0	1

 $\Phi_1 \oplus \Phi_0$ 

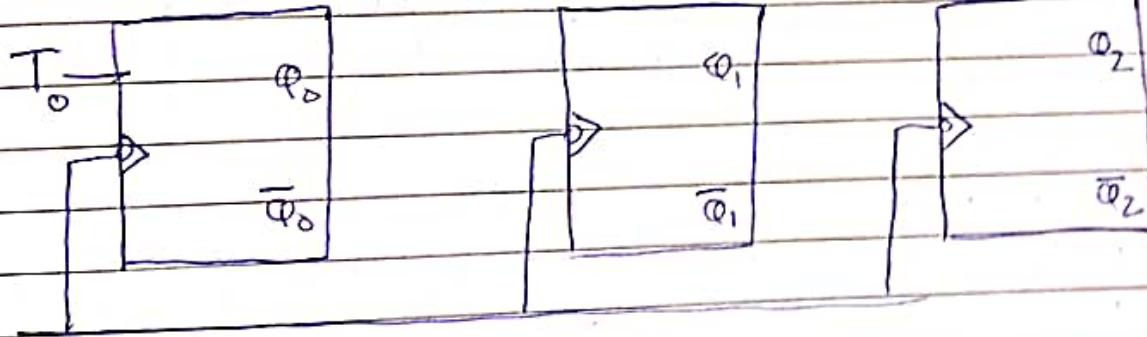
$\Phi_0$	0	1	1	1
1	0	2	1	0
$\Phi_1$	1	1	1	1
$\Phi_2$	1	1	1	1

$$T_2 = \Phi_2 \Phi_1 + \Phi_0 \Phi_1 + \Phi_0 \Phi_2$$

$$T_1 = \Phi_2 \Phi_1 + \bar{\Phi}_2 \Phi_0$$

$$T_0 = \bar{\Phi}_2 + \Phi_0 + \bar{\Phi}_1$$

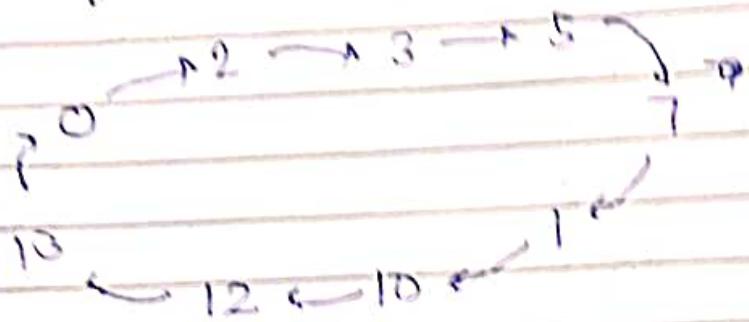
$\Phi_0$	0	1	1	1
1	1	2	1	0
$\Phi_1$	1	1	1	1
$\Phi_2$	1	1	1	1



Ques: Design an updown synchronous counter with MOD controller bit.

LW

Q) Design a counter with such a state diagram.



LW

1 Repeat above problem with all even numbered states are jumping for 3 and all odd numbered states are jumping for 12.

S-1

S-IV S-VI

					T	$Q_1 Q_0$
0 0 0 0					0	Q <sub>1</sub> Q <sub>0</sub>
0 0 0 1 0					40	-
0 0 1 1	1101	1100	0000	0011	1	1 1
0 1 0 1		1100	0010	0011	1	0 1
0 1 1 1			0000	0011	1	1 0
0 0 0 1						
1 0 1 0						

		$Q_3$	$Q_2$	$Q_1$	$Q_0$	$Q_{3+1}$	$Q_{2+1}$	$Q_1 T_2$	$T_3$	
0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	1	0
2	0	0	1	0	0	0	1	0	0	0
3	0	0	1	1	0	1	0	1	1	1
4	0	1	0	0	0	0	0	0	0	0
5	0	1	0	1	0	1	1	0	0	1
6	0	1	1	0	0	0	0	0	1	1
7	0	1	1	1	0	0	0	1	1	1

	$\Phi_3$	$\Phi_2$	$\Phi_1$	$\Phi_0$	$\Phi_3 \Phi_2 \Phi_1 \Phi_0$	$T_3$	$T_2$	$T_1$	$T_0$
8	1	0	0	0	0 0 0 0	1 0 0 0			
9	1	0	0	1	0 0 0 0	1 0 0 1			
10	1	0	1	0	1 1 0 0	0 1 1 0			
11	1	0	1	1	0 0 0 0	1 0 1 1			
12	1	1	0	0	1 1 0 1	0 0 0 1			
13	1	1	0	1	0 0 0 0	1 1 0 1			
14	1	1	1	0	0 0 0 0	1 1 1 0			
15	1	1	1	1	0 0 0 0	1 1 1 1			

 $T_3$  $T_2$  $\Phi_3 \Phi_2$ 

01	11	10
4	12	9
5	13	11
6	14	12
7	15	13

$\Phi_1 \Phi_0$	00	01	11	10
00	0	4	1	5
01	1	5	1	9
11	2	7	1	11
10	2	6	1	10

$\Phi_3 \Phi_2$	00	01	11	10
00	0	1	1	2
01	1	1	1	4
11	1	1	1	3
10	2	1	1	10

$$T_3 = \Phi_3 \Phi_0 + \Phi_3 \Phi_2' \Phi_1' + \Phi_3 \Phi_2 \Phi_1 + \Phi_2' \Phi_1' \Phi_0$$

01	11	10
4	12	8
5	13	19
6	14	11
7	15	17

$$T_2 = \Phi_2 \Phi_1 + \Phi_3 \Phi_1 \Phi_0' + \Phi_3' \Phi_2 \Phi_0' + \Phi_2' \Phi_1' \Phi_0 + \Phi_3 \Phi_2 \Phi_0$$

$$T_{21} = \Phi_1 \Phi_0 + \Phi_3 \Phi_1 + \Phi_3' \Phi_0 + \Phi_2 \Phi_1 + \Phi_2' \Phi_2' \Phi_1'$$

$$T_0 = \Phi_3 \Phi_0 + \Phi_3 \Phi_2 \Phi_1' + \Phi_3' \Phi_2' \Phi_1' + \Phi_3' \Phi_2' \Phi_0'$$

$Q_3 Q_2 Q_1 Q_0$

7 0 0 1 1

8 1 0 0 0

9 1 0 0 1

10 1 0 1 0

11 1 0 1 1

12 1 1 0 0

13 1 1 0 1

14 1 1 1 0

15 1 1 1 1

$Q_3 Q_2 Q_1 Q_0$

0 0 0 1

0 0 1 1

1 1 0 0

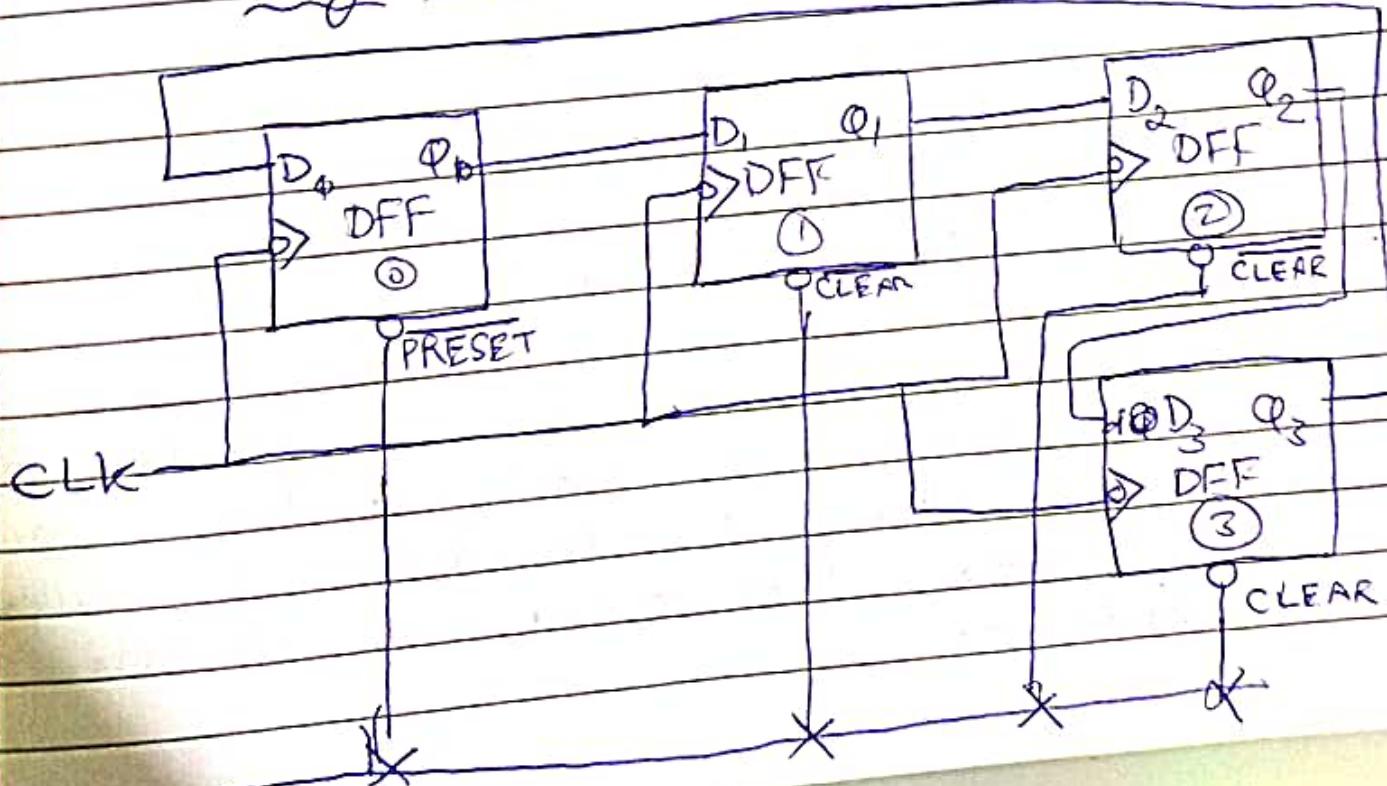
$T_3 T_2 T_1 T_0$

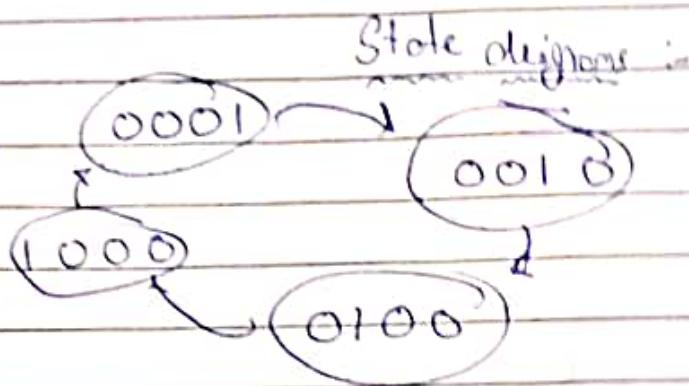
(Q) Repeat problem for the invalid states jumping  
 to  $\sim 0.75$  of the ~~avg~~ invalid state value.  
 the state  
 having  
 value

Shift Register Counter  
 (D flip flop based)

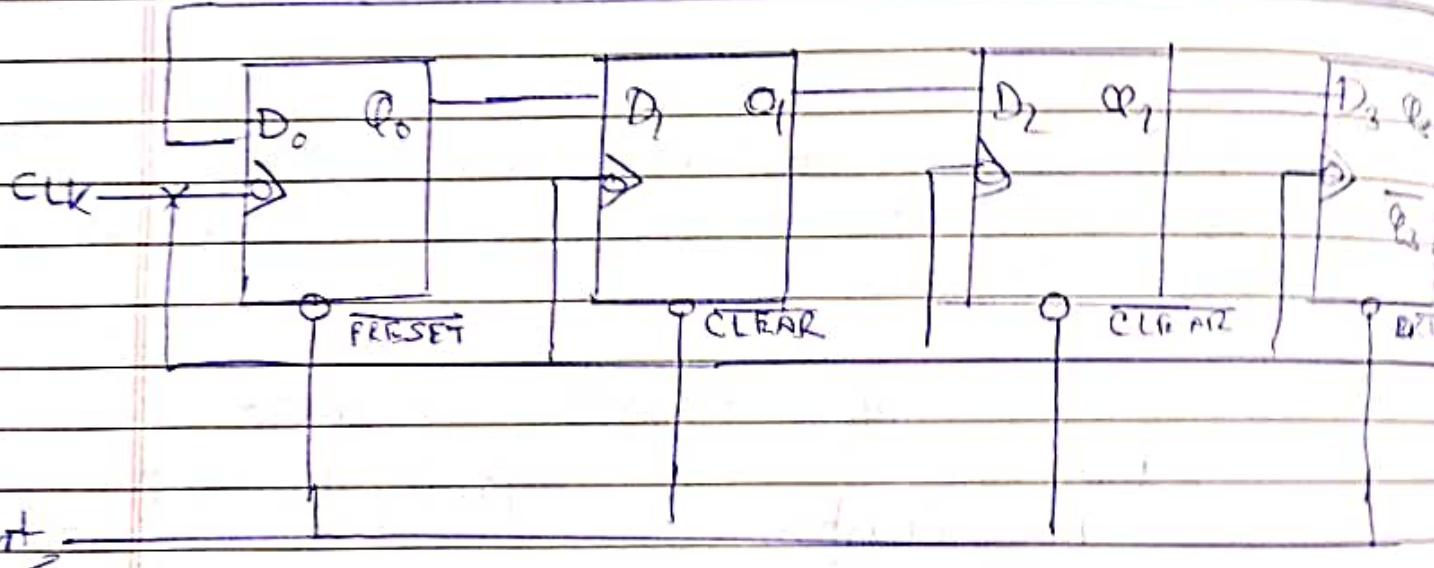
Ring

Ring Counter





Twisted State Ring Counter



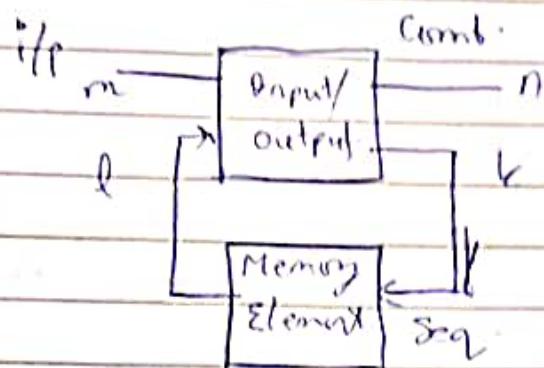
## FSM

### Finite State Machine :-

A FSM is an abstract model describing the synchronous machines. The behaviour of an FSM is described as a sequence of events that occur at discrete time instances.

These machines are called FSM because it has a fixed number of states.

To cg: The vending machines that provides different products are an example of FSM.



Therefore FSM can be of SISO, SIPO, MISO, MIPO type of FSM

There are two type FSM's possible.

- 1) Moore ckt.
- 2) Mealy ckt.

In Moore ckt's output depends only on the present state of flip flops while in Mealy ckt's , output depends on both present state of flip-flops and the inputs.

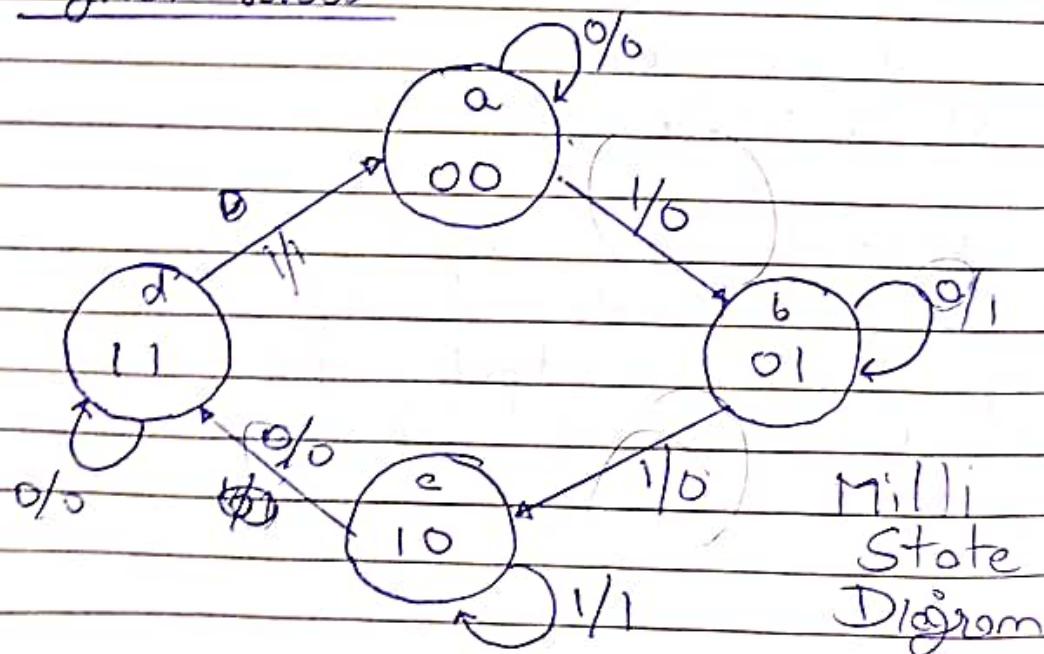
## Important terms of FSM

State diagram : It is a pictorial representation of states and their state relationship of FSM.

State Table : It is the tabular representation of state diagram.

State reduction : If redundant states are available in any state diagram then such states should be removed from the FSM. This helps to reduce the hardware requirement of FSM.

Examples of Moore & Meille ckt state diagram are given below :-



{ State }  
 [ Table ]

P.S

N.S

O/P

Input - ~~0 & 1~~ Z

0

Z=0

Z=1

Y=0

Y=1

a

c

b

0

0

b

b

c

1

0

c

d

d

0

1

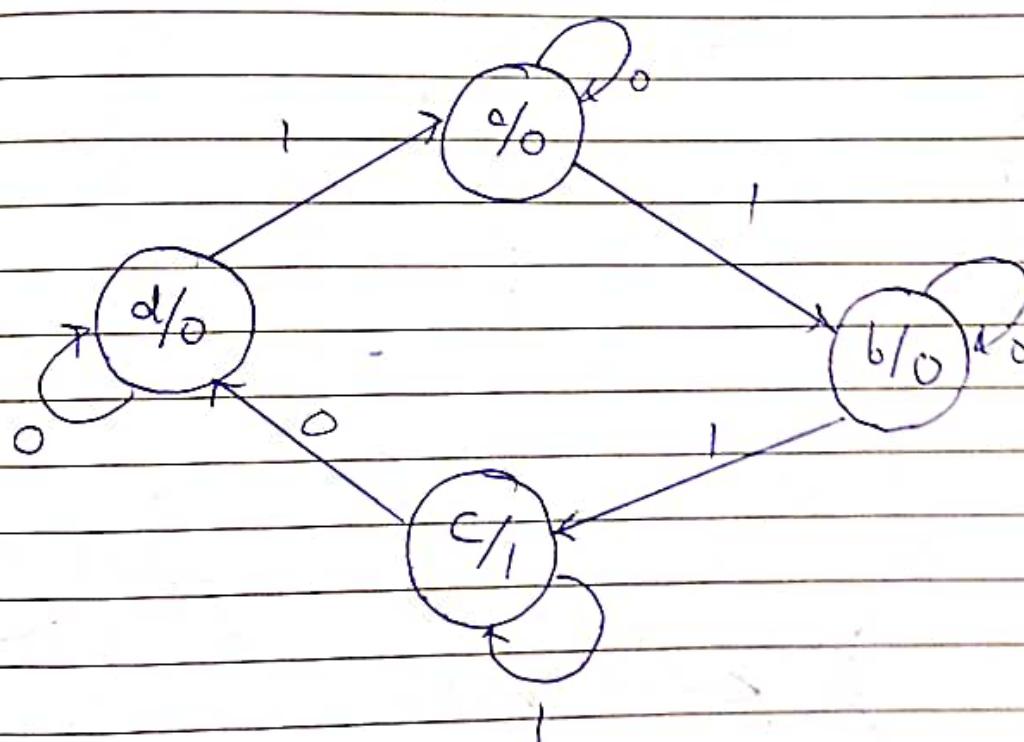
d

d

a

0

1



P.S

N.S

O/P

Z=0      Z=1

a

a

b

0

b

b

c

0

c

d

c

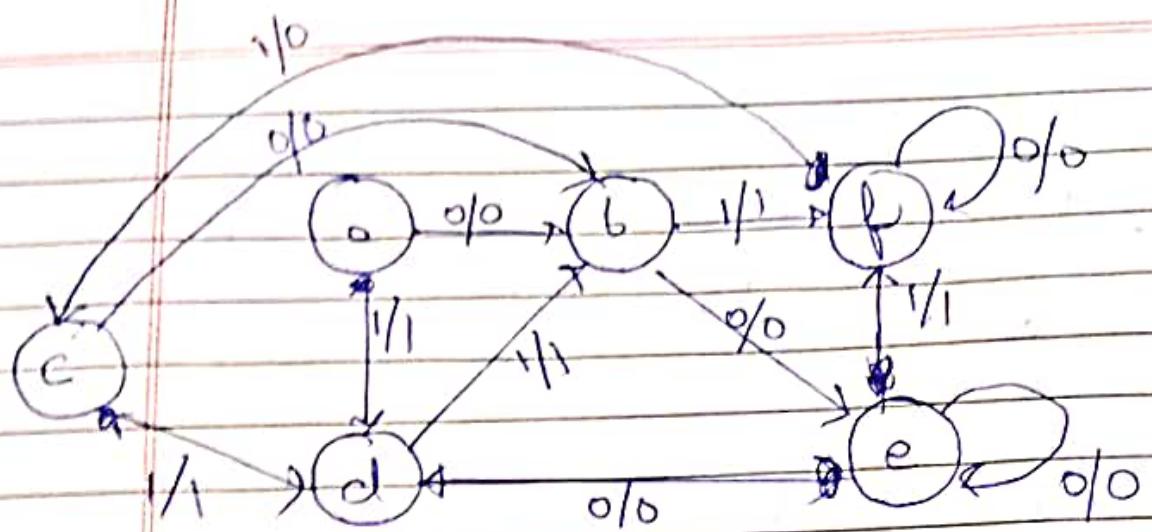
1

d

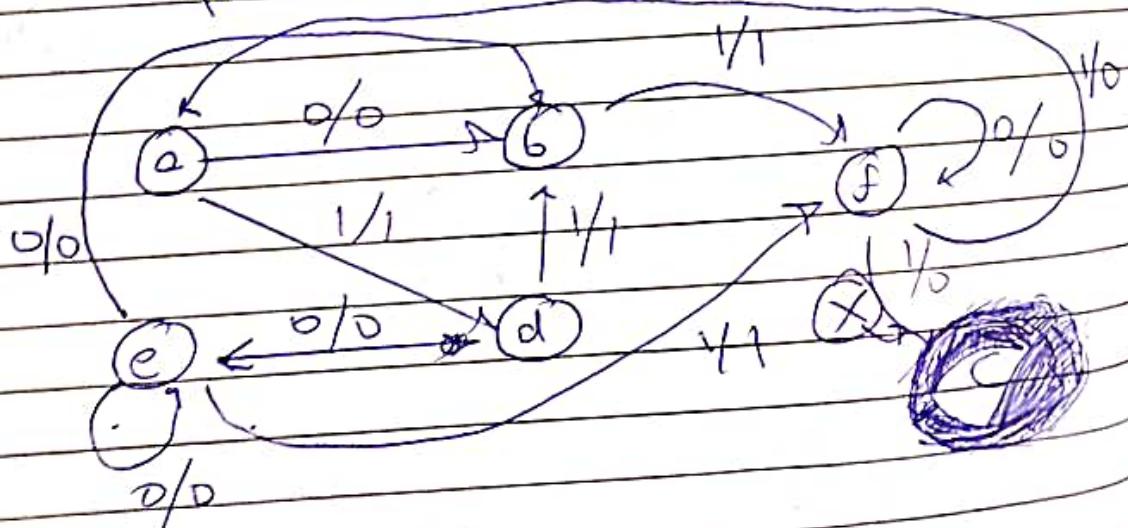
d

c

1



PS	NS	O/P
a	b	$y_2 = 0$
b	ab	$y_2 = 1$
c	b	$y_2 = 0$
d	a	$y_2 = 1$
e	e	$y_2 = 0$
f	f	$y_2 = 0$



Now transition  $a \rightarrow b$  with  $x=0$  produces output 0, similarly transition  $B \rightarrow E$  produces output 0 when  $x=0$ .

PS

NS

o/p

 $Z_0 = 0$  $Z = 1$  $Z > 0$  $Z = 1$ 

a

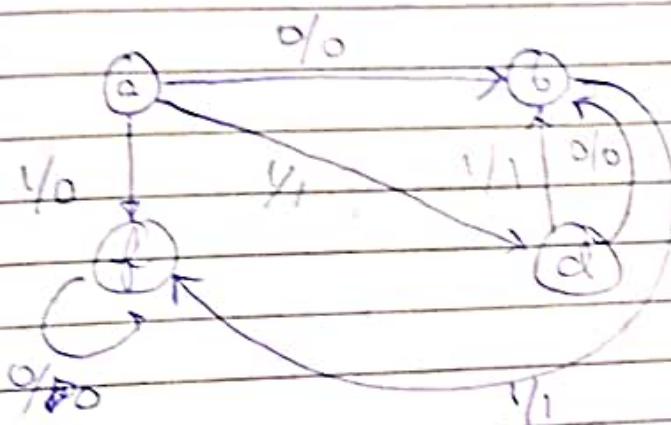
b

c

d

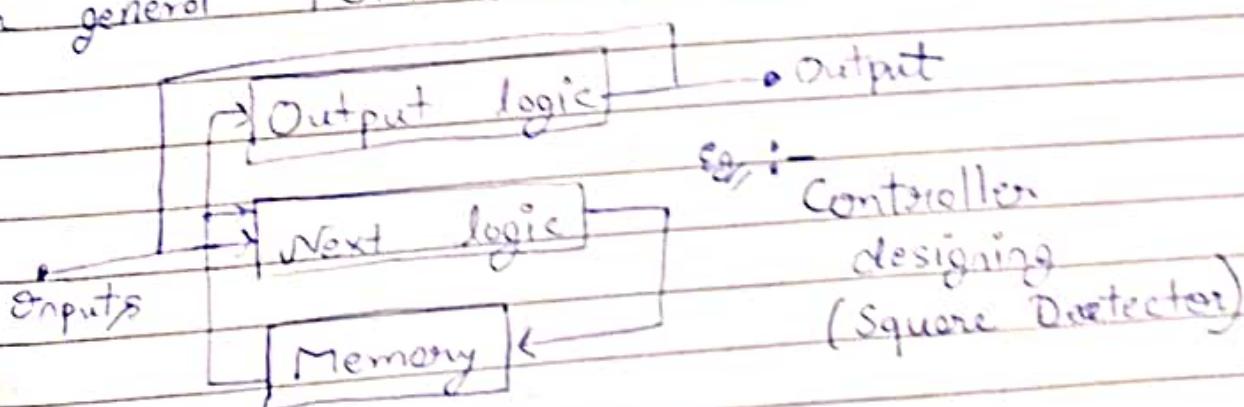
e

f



HW: Implement FSM of J-K Flip Flop

In general FSM can be drawn like this



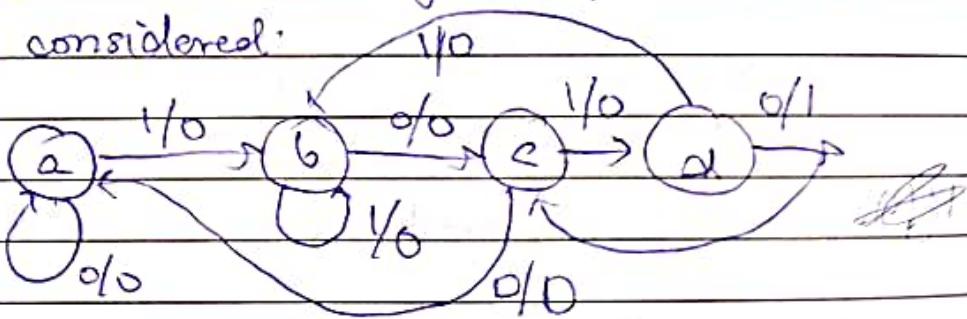
Ans

A sequence detector produces the output logic 1 when it detects the incoming bit sequence 1010.

Step 1: The controller is a sequence detector which produces logic 1 when it detects 1010 in the incoming bit string.

Step 2: Here overlapping of when bit pattern is detected output  $Z = 1$  else 0.

~~Step 3:~~ State diagram state table and its reduction. It is better to start the designing with optimum number of states. Here 4 states are considered.



State Table

PS	NS		O/P	
	$Z_{10}$	$Z_{21}$	$g_{10}$	$g_{21}$
a	a	b	0	0
b	c	b	0	0
c	a	d	0	0
d	c	b	1	0

State reductions are not required as minimum number of states are not considered.

### Step 3

State assignment and output table

PS	NS		O/P	
	$Z=0$	$Z=1$	$y=0$	$y=1$
a (00)	$s = 100$	b (01)	0	0
b (01)	c (10)	b (01)	0	0
c (10)	a (00)	d (11)	0	0
d (11)	c (10)	b (01)	1	0

### Step 4 :

Excitation table as per flip flop and K-Maps

D Q <sub>1</sub> ' Q <sub>0</sub> '	PS	Input	N.S	Input to Flip Flop		O/P
				D <sub>1</sub>	D <sub>0</sub>	
0 0	0 0	0 0	0 0	0	0	0 0
0 1	0 1	0 0	0 1	0	1	0 0
1 0	0 2	0 1	1 0	1	0	0 0
1 1	1 3	0 1	0 1	0	1	0 0
	4	1 0	0 0	0	0	0 0
	5	1 0	1 1	1	1	0 0
	6	1 1	1 0	1	0	0 1
	7	1 1	0 1	0	1	1 0

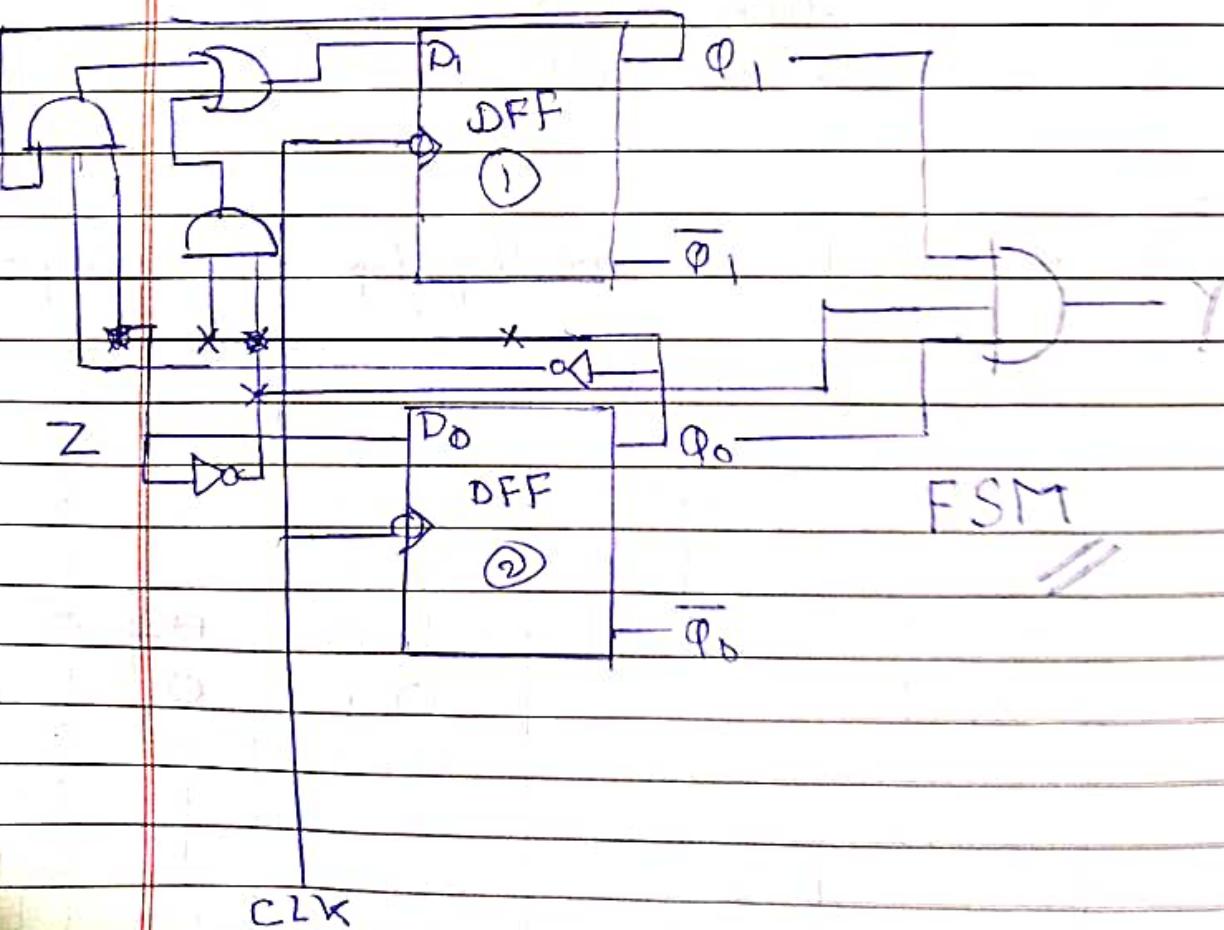
$\Phi_1 \Phi_0$	00	01	11	10
0	0	(1)	1	0
1	1	0	1	(1)

$$Y = D_1 = \Phi_0 Z' + \Phi_1 \Phi_0 Z$$

$$D_0 = Z$$

$$Y = \Phi_1 \Phi_0 Z'$$

Step 5: circuit diagram of FSM



HW Bcs.

Youtube

## Finite State Machine

Double circle → Final state

$$(\emptyset, \Sigma, q_0, F, \delta)$$

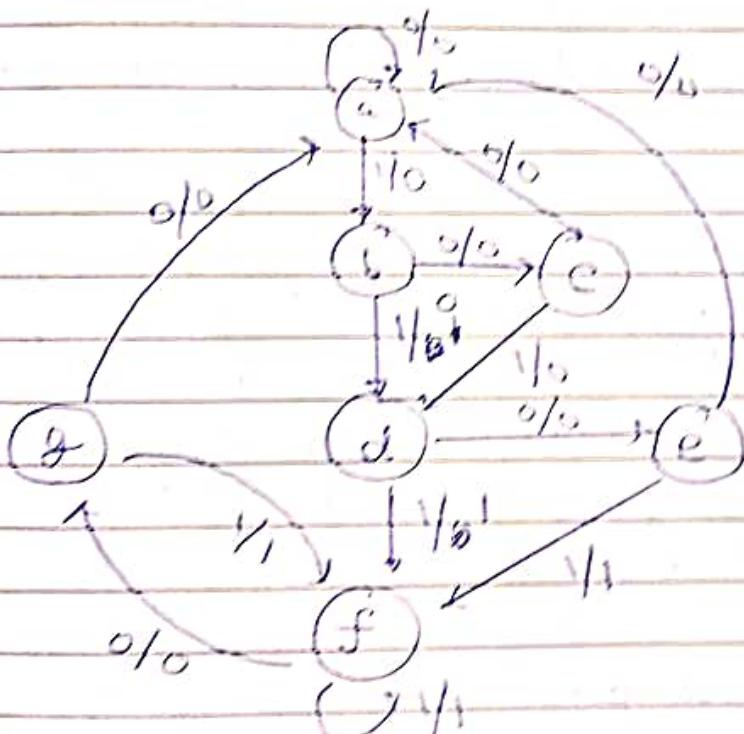
$\emptyset$  = finite set of all states

$\Sigma$  = inputs

$q_0$  = start state/initial state

$F$  = set of final state

$\delta$  = transition function



PS

NS

0/1/a/b

a a b c d e f

b c d e f g h

c d e f g h i j

d e f g h i j k l

e f g h i j k l m

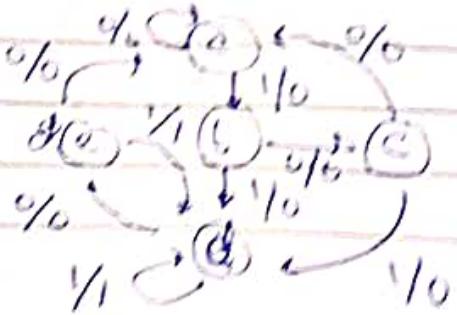
PS

NS

0/1/a/b

a a b c d e f  
b c d e f g h  
c d e f g h i j  
d e f g h i j k l  
e f g h i j k l m  
f g h i j k l m n  
g h i j k l m n o  
h i j k l m n o p  
i j k l m n o p q  
j k l m n o p q r  
k l m n o p q r s  
l m n o p q r s t  
m n o p q r s t u  
n o p q r s t u v  
o p q r s t u v w  
p q r s t u v w x  
q r s t u v w x y  
r s t u v w x y z  
s t u v w x y z u  
t u v w x y z u v  
u v w x y z u v w  
v w x y z u v w x  
w x y z u v w x y  
x y z u v w x y z  
y z u v w x y z u  
z u v w x y z u v

c=g, d=f,



HDL

- Hardware Description Language

L

VHDL

→ VHSIC

very high speed integrated circuits

Verilog

HDL stands for 'Hardware description language' and is used for describing the hardware on a computer. It is hard and difficult to understand. This is high level language which has syntaxes and sentences to describe the circuit. The most commonly used languages are VHDL & Verilog.

In verilog, most commonly 2 types of modelling are followed - ~~are~~ behavioural modelling — structural modelling

The behavioural modelling describes what a module can do. This modelling uses assignment statements, loops, if else, such kind of statements while structural modelling describes the components of the hardware. This requires interconnections of multiple components required in the circuit.

In Verilog, 2 major purposes i.e logic simulation and synthesis are being followed. During simulation inputs are applied to module and outputs are checked to verify the module. & During synthesis, the textual

description of a module is transformed into logic gates.  
In HDL, the HDL code represents the circuit  
and not end not a code which resembles the  
other programming language of a computer.

There are multiple levels of abstractions possible in  
VHDL like transistors.

Verilog - can describe a circuit model by using  
data flow modelling, behavioural modelling

### Gate level abstraction

Simulator is used to simulate the circuit  
virtually.

Data flow  $\rightarrow \{$  continuous assignment statement  $\}$

expressions :-  $i \cdot \overline{i} = 0 \rightarrow z$

Module contt  $(z, a, b)$

i/i inputs  $a, b;$   
o/o output  $z;$

assign  $z = a \& b;$

endmodule ;

Module ~~half~~ adder ( $\oplus$  sum, carry, c, b)

input a, b;

output sum, carry;

assign sum =  $a \wedge b;$

assign carry =  $a \& b;$

endmodule;

$$\textcircled{1} \quad f(a, b, c, d) = \sum (0, 1, 3, 5, 7, 8, 9, 10, 12, 13)$$

module full-adder(sum, carry, a, b, cin)

output sum, carry;  
input a, b, cin;

assign ~~sum~~<sup>carry</sup> =  $(a \& b) + (a \& \text{cin}) + (b \& \text{cin})$ ;  
assign sum =  $a ^ b ^ \text{cin}$ ;

end module

SOP & POS

c — X — — X — — X  
note       $2^3 b \rightarrow 00$        $4^1 H \rightarrow AB12$

( $\frac{2^3}{4^1}$ )  $3^1$        $4^1 b \rightarrow 0000$

[Decimal numbers  
can be represented  
normally]

$2^3 b 00$       } Representation  
 $4^1 b 0000$       }  
 $4^1 H AB12$

→ Write Verilog code for function f  
in data flow style

$$P(A, B, C, D) = \sum_{13} (0, 1, 2, 3, 5, 7, 8, 9, 10, 12)$$

		AB	00	01	11	10
		CD	00	01	11	10
A	B	00	1	4	12	1
		01	1	13	13	1
B	C	11	13	1	15	11
		10	1	6	14	10

$$F = \bar{A}\bar{B}' + A'D + A'C' + \cancel{ABC} + AB\bar{D}'$$

module exp

<del>CD</del>	<del>00</del>	<del>01</del>	<del>11</del>	<del>10</del>
<del>00</del>	<del>0</del>	<del>1</del>	<del>1</del>	<del>0</del>
<del>01</del>	<del>1</del>	<del>0</del>	<del>0</del>	<del>1</del>
<del>11</del>	<del>1</del>	<del>1</del>	<del>0</del>	<del>0</del>
<del>10</del>	<del>0</del>	<del>1</del>	<del>1</del>	<del>1</del>

$$F = A' \otimes D + A C' \\ + B'D'$$

<del>CD</del>	<del>00</del>	<del>01</del>	<del>11</del>	<del>10</del>
<del>00</del>	<del>0</del>	<del>1</del>	<del>1</del>	<del>0</del>
<del>01</del>	<del>1</del>	<del>0</del>	<del>0</del>	<del>1</del>
<del>11</del>	<del>1</del>	<del>1</del>	<del>0</del>	<del>0</del>
<del>10</del>	<del>0</del>	<del>1</del>	<del>1</del>	<del>1</del>

$$\pi ( = 5, 7, 9, 10, 11, 13, 14, 15 )$$

$$F = \cancel{B'C} + (B' + D')(A' + D') \\ (B' + C')(A' + C')$$

module  $\text{exp}(f, A, B, C, D)$

output  $f$ ;  
input  $A, B, C, D$ ;

begin  
 $f = \cancel{A}((\sim A \wedge D) \Rightarrow ((A \wedge (\sim C)) \wedge ((\sim B) \wedge (\sim D)))$ ;

endmodule;

CLR Sequential  $\Rightarrow$  Behavioral

Behavioral

} }  $\downarrow$

Procedural assigned  
statements

\* always  
seq { }

J-K flip flop

$(Q_{n+1}, \bar{Q}_{n+1})$	$J$	$K$	$Q_n$	$T$	$\bar{T}$	$C_{in}$	$C_{out}$
$(0, 0)$	0	0	0	0	1	0	0
$(1, 1)$	0	0	1	0	1	1	1
$(0, 1)$	1	0	0	1	0	0	0
$(1, 0)$	0	1	1	1	0	1	1
$(Q_n, \bar{Q}_n)$							
$D$	$Q_n$						
$C_{in}$	$\rightarrow$	$\bar{Q}_n$					

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

module DFF1 ( D, clk, Q, Qb )

input D, clk;

output Q, Qb;

reg Q;

always @ ( posedge ) clk )

Q = D;

Qb =  $\overline{D}$ ;

endmodule ;

module JKFF1 ( J, K, clk, Q, Qb )

input J, K, clk;

output Q, Qb;

reg Q;

always @ ( posedge clk or posedge RST )

begin  
case { J, K } Q = 0;

2'b00 : Q = 0; 1'b0;

2'b01 : Q = 0; 1'b0;

2'b10 : Q = 1; 1'b0;

2'b11 : Q = Qb; 1/ or ~Q)

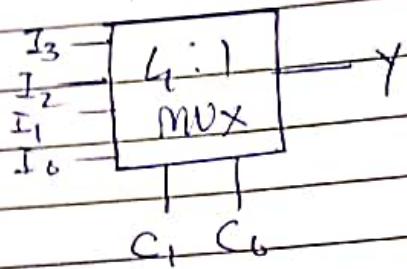
for triggering  
flip flop

endcase;

endcase; end;

endmodule;

Mux 4:1 using behavioural style.



module mux\_1 (Y, C1, C0, I0, I1, I2, I3)

input I0, I1, I2, I3;

output C1, C0;

begin

always @ (I0, I1, I2, I3, C1, C0)

~~Y = case { C1, C0 }~~

2'6 00 : Y = I0;

2'6 01 : Y = I1;

2'6 10 : Y = I2;

2'6 11 : Y = I3;

~~end~~

endcase;

endmodule;