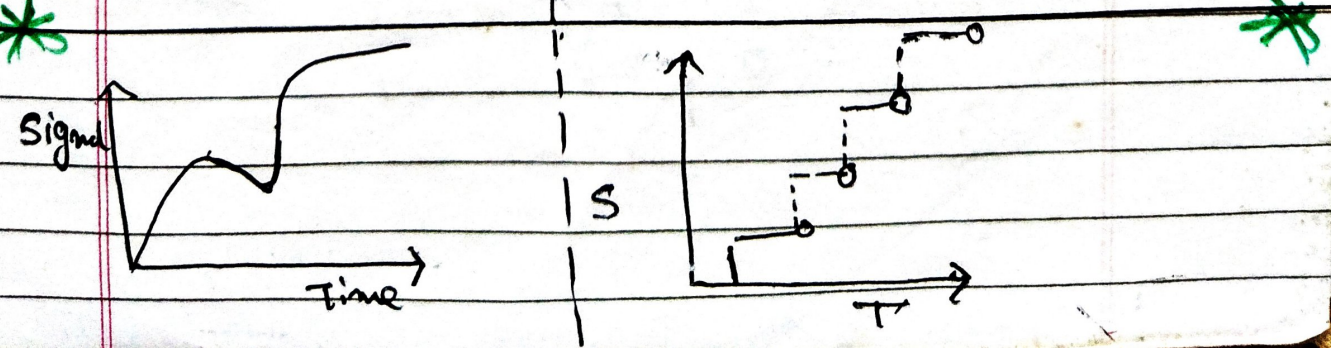
DIGITAL TECHNIQUS

Page

DIGITAL ELECTRONICS

-: LOGIC GATES :-

	Analog signal	Digital signal
1-	An Analog signal is a continous signal that varies over time	1- A digital signal is a discrete signal that has a finite number of levels
2-	It can take any value within a give range.	2- usually represented as binary (0 and 1)
3-	Represented by a sine wave	3- it is more resistant to noise.
4-	Example - sound waves and the transmitted by old radios	4- Commonly used in computers, smartphone and others digital devices



FREEMIND

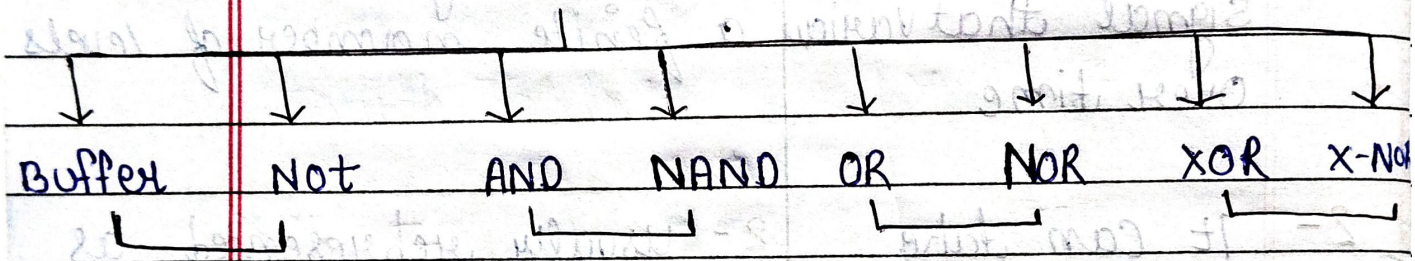
Date

Page

GATES :-

Logic gate is a electronic circuit which have one or more than one input but having only one output and both are digital

Types of Gate :-

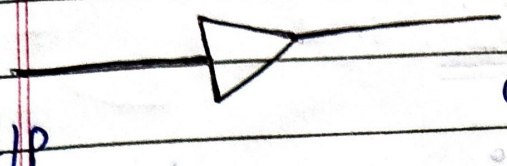


* They all are opposite each other

Logic Gates

Basic gates	Universal gates	Arithmetic Gates
→ AND	→ NAND	→ XOR
→ OR	→ NOR	→ X-NOR
→ NOT		

① Buffer Gate



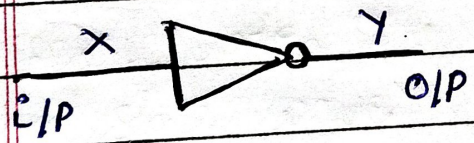
Definition: Output will be same as input

Equation: $Y = X$

Truth table:

X	Y
0	0
1	1

② NOT GATE



Definition: Output will inverd of input

Equtions: $Y = \bar{X}$

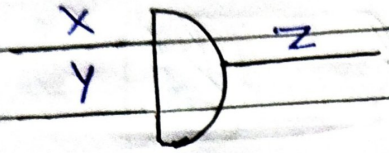
Truth Table:-

X	Y
0	1
1	0

Date

Page

(3) AND GATE



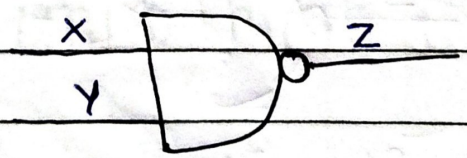
Definition :- if any input is zero that output equal to zero

Equation :- $Z = X \cdot Y$

Truth Table

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

(4) NAND GATE



Definition :- if any input is 0 Output is 1

Equtions :- $Z = (X \cdot Y)'$

$Z = X' + Y'$

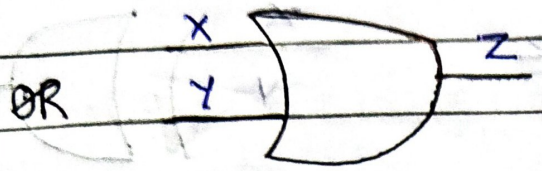
Truth Table :

X	Y	Z
0	0	1

Date

Page

(5) OR GATE



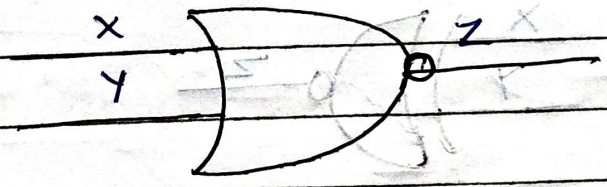
Definition :- if any input is one that output is 1

Equation :- $Z = X + Y$

Truth Table :-

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

(6) NOR GATE



Definition :- If any input is 1 than output is zero

Equation :- $Z = \overline{X + Y}$

Truth Table :-

X	Y	Z
0	0	1
0	1	0
1	1	0

FREEMIND

Date

Page

(7) XOR GATE

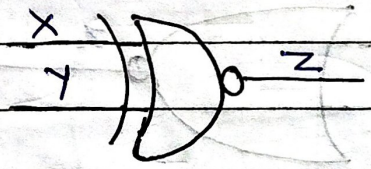
Definition :- If odd No. of input is 1 than output is 1

Equation :- $Z = X \cdot \overline{Y} + \overline{X} \cdot Y = X \oplus Y$

Truth Table :-

X	Y	Z
0	0	0
1	0	1
0	1	1
1	1	0

(8) XNOR GATE



Definition :- If even no of input is 1 than output is 0

Equation :- $(A \cdot B) + (\overline{A} \cdot \overline{B}) = \overline{A \oplus B}$

Truth Table

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

Date

Page

Equation :-

$$z = (x'.y + x.y')$$
$$= [(x')' + y'] . [x' + (y')']$$
$$= x.x' + x.y + x'.y' + y'.y$$

1 $\longrightarrow$ $[z = x.y + x'y']$

2 $\longrightarrow$ $[z = x \odot y]$

Universal Gate :-

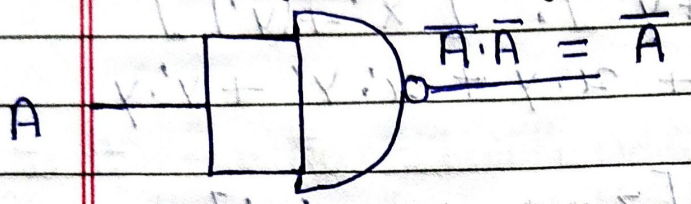
A universal gate in digital logic is a type of logic gate that can be used to implement any other type, such as AND, OR, NOT

There are 2 main types of universal gate

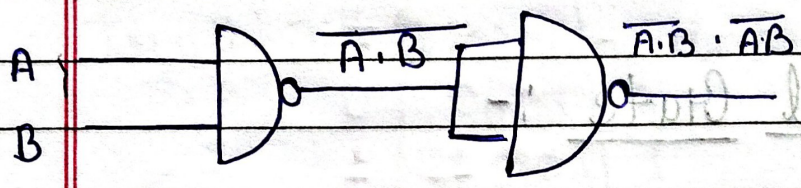
1. NAND
2. NOR

(1) NAND GATE AS A UNIVERSAL GATE

i) Not gate using NAND gate :-

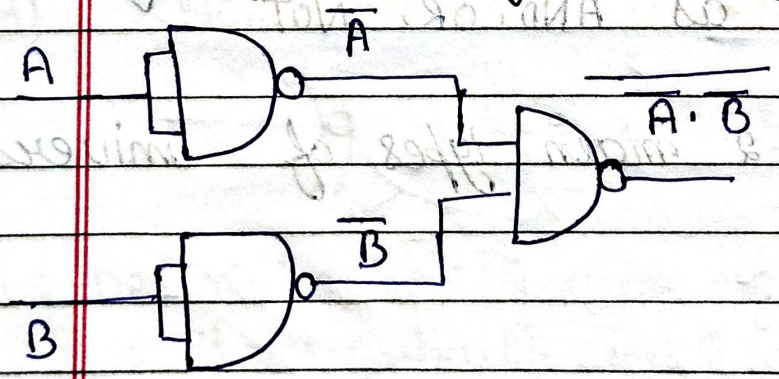


ii) AND gate using NAND gate



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

iii) OR gate using NAND gate



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

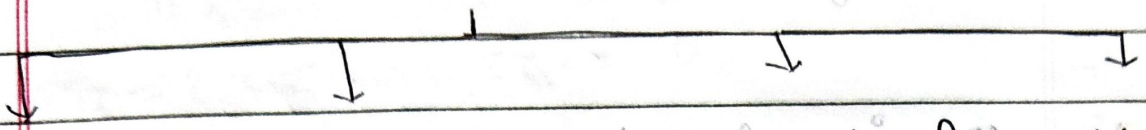
DeMorgen Thorem

FREEMIND

Date

Page

NUMBER SYSTEM

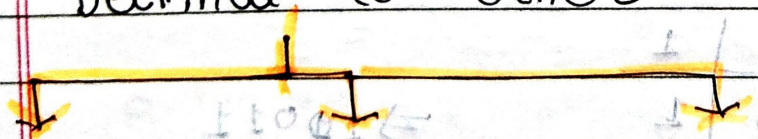


Binary Octal Decimal Hexadeci

	Number system	Base	Range
1→	Binary	2	(0,1) ⇒
2→	Octal	8	(0,7) ⇒ 0,1,2,3,4,5,6
3→	Decimal	10	(0,9) ⇒ 0,1,2,3,4,5,6,7,8,9
4→	Hexadecimal	16	(0,15) ⇒ 0,1,2,3,4,5,6,7,8,9
			*A = 10, C = 12, E = 14 B = 11, D = 13, F = 15

Conversion :-

1) Decimal to Other



Binary Octal Hexadecimal

i) Binary to Octal

1) Decimal to Binary

(i) 19.35

(1)

2	19.35	1
2	9	1
2	4	0
2	2	0
	1	

⇒ 10011

0.35 × 2 = 0.70 ⇒ 0
0.70 × 2 = 1.40 ⇒ 1
0.40 × 2 = 0.80 ⇒ 0

(19.35)10 → (10011.010)2

(2) (19.7)10

2	19	1
2	9	1
2	4	0
2	2	0
	1	

⇒ 10011

0.7 × 2 = 0.14 = 0
0.14 × 2 = 0.28 = 0
0.28 × 2 = 0.56 = 0

(19.7)10 = (10011.00)2

2) Decimal to Octal

(a) 58.5

8	58	2
8	7	7
	0	

⇒ 72

0.58 × 8 = 4.64

(58.5)10 = (72.4)8

(b) 19.35

8	19.35	3
8	2	2
	0	

⇒ 23

0.35 × 8 = 2.80 ⇒ 2
0.80 × 8 = 6.40 ⇒ 6

(19.35)10 = (23.26)8

3) Decimal to Hexadecimal

① 58.5

16	58	10
16	3	3
	0	

⇒ 3A

0.5 × 16 = 8

$(58.5)_{10} = (3A.8)_{16}$

② 19.35

16	19.35	3
16	1	1 ↑
	0	

⇒ 13

0.35 × 16 = 5.6 = 5

$(19.35)_{10} = (13.5)_{16}$

Binary

Convert Octal, Decimal and Hexadecimal

1) $(29)_{10} = (\ ?\ )_{2/8/16}$

(a) $(29)_{10} = (?)_2$

2	29	1
2	14	0
2	7	1
2	3	1 ↑
2	1	1
	0	

⇒ 11101

$(29)_{10} = (11101)_2$

(b) $(29)_{10} = (?)_8$

8	29	5
8	3	3 ↑
	0	

$(29)_{10} = (35)_8$

(c)

16	29	13
16	1	1 ↑
	0	

⇒ (1D)

$(29)_{10} = (1D)_{16}$

* Binary, Octal and Hexadecimal to Decimal.

✓ Binary
✓ Octal → Decimal
✓ Hexadecimal

1) Binary Convert to Decimal

(a) $(1000)_2 \rightarrow (?)_{10}$

8 4 2 1
1 0 0 0 $= (8)_{10}$

(b) $(111001)_2$

32 16 8 4 2 1
1 1 1 0 0 1

$32 + 16 + 8 + 1 = (57)_{10}$

2) Binary Convert to Octal

① $(110\ 110)_2$

6 6

$(110110)_2 = (66)_8$

② $(010\ 110\ 010)_2 = (132)_8$

③ $111\ 000\ 101 = (705)_8$

(3) Binary Convert to Hexadecimal

① $(00\ 1\ 1\ 0\ 1\ 1\ 0)_2$

$\Rightarrow (36)_{16}$

② $(1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1)_2 = 1125 \Rightarrow (1C5)_{16}$

③ $(0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 1)_2 = (25)_{16}$

1) Octal to Binary

① $(11.523)_8 = (\ ?\ )_2$

$(11\overleftarrow{5}.\overrightarrow{23}) \rightarrow (\ )_{10} \rightarrow (\ )_2$

$1\times8^2 + 1\times8^1 + 5\times8^0$
$64 + 8 + 5$
$\Rightarrow 77$

$2\times8^{-1} + 3\times8^{-2}$
$\frac{2}{8} + \frac{3}{64}$
$0.25 + 0.96$
$\Rightarrow 0.296$

$\Rightarrow (77.296)_{10} \rightarrow (1001101.01)_2$

2	77	1
2	38	0
2	19	1
2	9	1
2	4	0
2	2	0
	1	0

$\Rightarrow 1001101$

$0.296 \times 2 = 0.592 \Rightarrow 0$
$0.592 \times 2 \Rightarrow 1.184 \Rightarrow 1$

2) Octal to Decimal

① $(1057.06)_8 \rightarrow (\quad)_{10}$

$1\times8^3 + 0\times8^2 + 5\times(8)^1 + 7\times8^0$

$512 + 40 + 7$

$\Rightarrow 559$

$0\times(8)^{-1} + 6\times(8)^{-2}$

$\Rightarrow \frac{6}{64} = 0.937$

$\Rightarrow (1057.06)_8 = (559.937)_{10}$

3) Octal to Hexadecimal

① $(651.124)_8 \rightarrow (\quad)_2 \rightarrow (\quad)_{16}$

4 2 1	4 2 1	4 2 1
0 0 1	0 1 0	1 0 0

4 2 1	4 2 1	4 2 1
1 1 0	1 0 1	0 0 1

$(110\ 1010\ 0.001\ 010\ 100)_2 \rightarrow (\ ?\ )_{16}$

$(1A9.2A)_{16}$

1) Hexadecimal to Binary

① $(15)_{16} \rightarrow (\quad)_2$

8 4 2 1	8 4 2 1
0 0 0 1	0 1 0 1

$(15)_{16} = (00010101)_2$

② $(49)_{16} \rightarrow (\quad)_2$

8 4 2 1	8 4 2 1
0 1 0 0	1 0 0 1

$(49)_{16} = (0100\ 1001)_2$

③ $(A5C)_{16} \rightarrow (\quad)_2$

8 4 2 1	8 4 2 1	8 4 2 1
1 0 1 0	0 1 0 1	1 1 0 0

$(A5C)_{16} = (1010\ 0101\ 1100)_2$

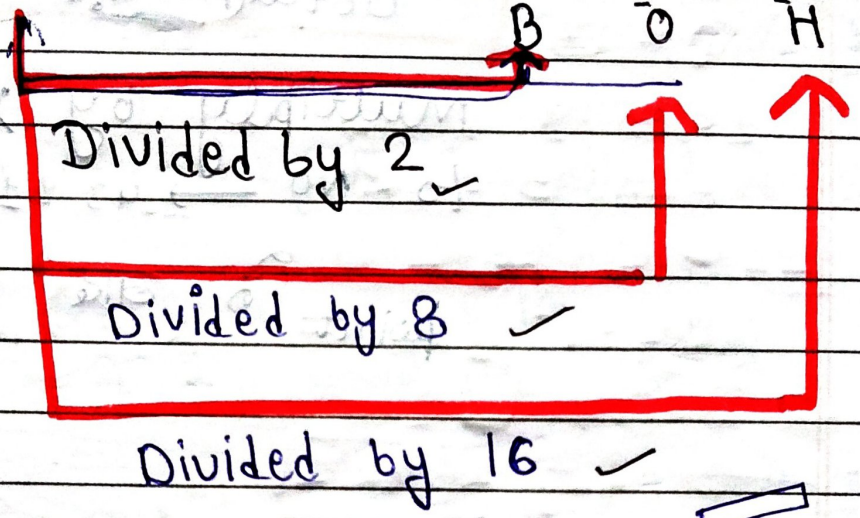
Date

Page

Important NOTES

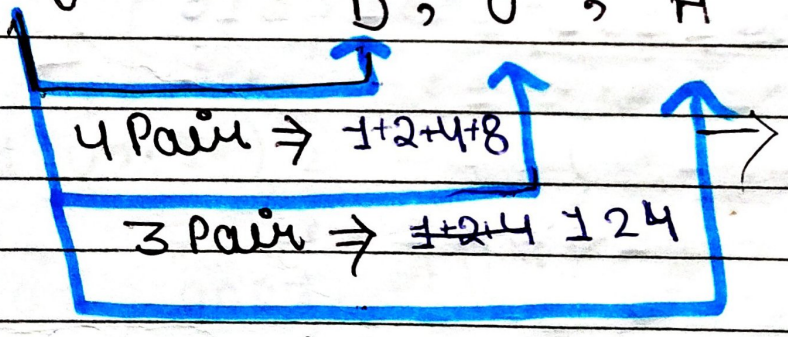
* **Conversion** :- Decimal $( )_{10} \rightarrow$ 8, 16, 2

1) Decimal Convert ⇒ 2, 8, 16
[10] - B O H

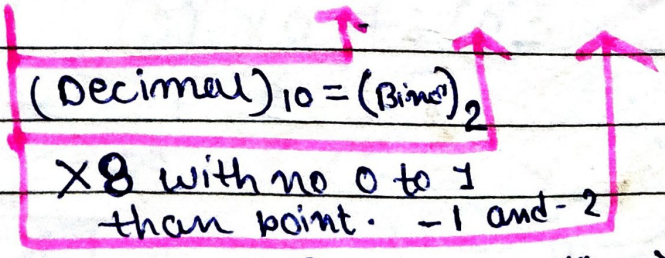


$(Binary)_2 =$ 0, 8, 16

2) Binary ⇒ 10, 8, 16
[2] D, O, H



3) Octal ⇒ 2, 10, 16
[8]



3 pair per NO → 124 (Binary)2

4pair

FREEMIND

Date

Page

* 4) Hexadecimal (16) ⇒ 2, 8, 10
B, O, D

pair per Number
1 2 4 8

pair per No ⇒ 8 4 2 1
Octal(pair3)$_8$ ← (Binary)$_2$

multiply by x16 according to No → 4 3 2 1 0

point के बाद — 1x16 x - 2x16

Binary Addition

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 0$ with 1 Carry

(1)

$$\begin{array}{r} 1\ 0\ 1\ 1\ 1 \\ +\ 1\ 1\ 0\ 1\ 1 \\ \hline 1\ 1\ 0\ 0\ 1\ 0 \end{array}$$

(2)

$$\begin{array}{r} 1\ 0\ 1\ 1 \\ +\ 0\ 1\ 1\ 0 \\ +\ 1\ 0\ 1\ 1 \\ \hline 1\ 1\ 0\ 1\ 0\ 0 \end{array}$$

(3)

$$\begin{array}{r} 1\ 1\ 1\ 0 \\ +\ 1\ 0\ 1\ 1 \\ \hline 1\ 0\ 1\ 0\ 0\ 1 \end{array}$$

(4)

$$\begin{array}{r} 1\ 0\ 1\ 0 \\ +\ 1\ 0\ 0\ 1 \\ \hline 1\ 0\ 0\ 1\ 1 \end{array}$$

FREEMIND

Date

Page

Binary Subtraction :-

0 - 0 = 0

1 - 0 = 1

1 - 1 = 0

0 - 1 = 1 with borrow

1→

	0	(10)→2			(10)
	1	1	0	1	16
-	1	0	1	1	- 8
	0	0	1	0	8

2→

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

0 - 1 = 1

1 borrow

3→

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

4→

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

Date

Page

Octal

* ~~Decimal~~ Addition :-

1 1

1) (5 6 7)8 ⇒ (0 - 7)

+ (2 4 3)8

1 0 3 2

7 + 3 = 10

$$8\overline{)10}\quad 1,\ 2$$

6 + 4 + 1 = 11

$$8\overline{)11}\quad 1,\ 3$$

5 + 2 + 1 = 8

$$8\overline{)8}\quad 1,\ 0$$

1 1 1 1

2) (3 2 7 . 5 4)8

+ (6 6 5 . 3 7)8

(1 2 1 5 1 3)8

$$8\overline{)11}\quad 1,\ 3$$

$$8\overline{)9}\quad 1,\ 1$$

$$8\overline{)13}\quad 1,\ 5$$

$$8\overline{)9}\quad 1,\ 1$$

$$8\overline{)10}\quad 1,\ 2$$

FREEMIND

Date

Page

Octal Subtraction :-

$10+3 = 13-4 = 8$

1) $\begin{array}{r} 7\ 4\ 3 \\ -\ 5\ 6\ 4 \\ \hline 1\ 5\ 7 \end{array}$

2) $\begin{array}{r} 8 \\ 2\ 5\ 7 \\ -\ 1\ 6\ 5 \\ \hline 0\ 7\ 2 \end{array}$

Base = 8

Borrow = 8

3) $\begin{array}{r} 7\ 2\ 5 \\ -\ 4\ 6\ 1 \\ \hline 2\ 4\ 4 \end{array}$

4) $\begin{array}{r} 5\ 3\ 2 \\ -\ 2\ 5\ 7 \\ \hline 2\ 5\ 3 \end{array}$

5) $\begin{array}{r} 1\ 0\ 0\ 0 \\ -\ 7\ 7\ 7 \\ \hline 0\ 1\ 1\ 1 \end{array}$

$\begin{array}{r} 7\ 1\ 4\ 3 \\ -\ 5\ 6\ 4 \\ \hline 1\ 5\ 7 \end{array}$

FREEMIND

Date

Page

Hexadecimal Addition :-

1) $(A65)_{16} + (25B)_{16}$

```
   1
   A 6 5
+  2 5 B
-------
   C C 0
```

A	B	C	D	E	F
10	11	12	13	14	15

```
   1
16 | 16
   16
   ---
    0
```

2)

```
   1
   A 9
+  4 7
-------
   F 0
```

```
   1
16 | 16
   16
   ---
    0
```

3)

```
   1
   2 C 3
   5 B 8
---------
   8 7 B
```

```
    1
16 | 23
   16
   --
    7
```

4)

```
   1 1
   2 C B
   5 B 8
---------
   8 8 3
```

```
    1
16 | 19
   16
   --
    3
```

```
    1
16 | 24
   16
   --
    8
```

FREEMIND

Date

Page

Hexdecimal subtraction :-

1) 9 A 5
− 8 B 4
0 F 1

Base = 16
Borrow = 16

2) 9 F 3 → 16
− 4 B 7
5 3 C

16 | 21 (1
16
5

16 | 18 (1
16
2

2C
11
15
C

14
11
B

Page

BCD :-

Binary Coded Decimal :-

Base = 10

0 to 9

1) 1 43

↓ ↓ ↓

8 4 2 1 8 4 2 1 8 4 2 1

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

⇒ (000101000011)

BCD

Binary Coded Decimal

FREEMIND

Date

Page

Gray code to binary conversion

1) 1 1 0 1

1 0 0 1

2) 1 1 1 1

1 0 1 0

3) 1 0 1 1

1 1 0 1

3) 1 0 1 1

1 1 0 1

3) 1 0 1 1

1 0 1

3) 1 0 1 1

⇒ 1 1 0 1

1 1 0 1

1 0 1 0

1 1 1 1

↓

1 0 1 0

X-OR

1 + 0 = 1

0 + 1 = 1

1 + 1 = 0

0 + 0 = 0

1 0 1 1 02

Date

Page

IMPORTANT

a) $x + 0 = 0$

b) $x + x' = 1$

c) $x + x = x$

d) $x + 1 = 1$

e) $(x')' = x$ $(x+y) = y + x$

f) $x(y+z) = xy + xz$

g) $(x+y)' = x' \cdot y'$

h) $x + xy = x$

$\overline{x \cdot y} = \overline{x} + \overline{y}$

---

a) $x \cdot 1 = x$

b) $x \cdot x' = 0$

c) $x \cdot x = x$

d) $x \cdot 0 = 0$

e) $xy = yx$

f) $x + y = (x+y)\ (x+z)$

g) $(x \cdot y)' = x' + y'$

h) $x(x+y) = x$

i) $\overline{xy} = \overline{x} \cdot \overline{y}$

FREEMIND

Date

Page

1) $x \cdot x + x \cdot z + x \cdot y + yz$

$x(1 + z + y) + yz$

$x \cdot 1 + yz$

$(x + yz)$

2) $x(x' + y)$

$xx' + xy$

$0 + xy$

3) $x'y' + xy + x'y$

$x'y' + y(x + x')$

$x'y' + y \times 1$

$x'y' + y$

4) $(x+y)(x+y')$

$x(x+y') + y(x+y')$

$x \cdot x + xy' + yx + yy'$

$x + xy' + yx + 0$

$x(y' + y)$

$\Rightarrow x$

19/09/24

Min Term :-

1) 2 Variable

x	y	Min
0	0	$x' \cdot y'$
0	1	$x' \cdot y$
1	0	$x \cdot y'$
1	1	$x \cdot y$

2) 3 Variable :-

x	y	z	Min
0	0	0	$x' \cdot y' \cdot z'$
0	0	1	$x' \cdot y' \cdot z$
0	1	0	$x' y z'$
0	1	1	$x' y z$
1	0	0	$x y' z'$
1	0	1	$x y' z$
1	1	0	$x y z'$
1	1	1	$x y z$

Max Term :-

No of Variable n: $2^n = 2^3 = 8$

1) 2 Variable

X	Y	Max
0	0	$X+Y$
0	1	$X+Y'$
1	0	$X'+Y$
1	1	$X'+Y'$

Min → SOP: $0 = X'$, $1 = X$

Max → POS: $1 = X'$, $0 = X$

(Min)

1) Convert SOP to POS expression

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

$= 000 + 010 + 011 + 101 + 111$

$Y = m_0 + m_2 + m_3 + m_5 + m_7$

$F = \Sigma (0, 2, 3, 5, 7)$ Min Term

$F = \pi (1, 4, 6) \rightarrow$ Max Term
(001, 100, 110)

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

K. MAP

Karnaugh map
(कार्नो)

K- map is a graphical tool used in digital logic design to simplify Boolean expression

Whdt is the use of k map
1) logic circuit simplification
2) Cost Effective circuit
3) faster Design process

Boolean expression ⇒ को सरल
circuit design or analyze करने
के लिए होता है।

Boolean expression is the language to define logic gates and circuits work.

FREEMIND
Date
Page

2 Variable :-

$2^n = 2^2 = 4$ $n \Rightarrow$ no of Variable

A \ B	0	1
0	00	01
1	10	11

Type - 1

A \ B	0	1
0	0	1
1	2	3

Type - 2

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

Type - 3

* K Map based on three types of Input Values (0, 1, don't care)
* K map is based on Gray code

3- Variable K Map

$2^n = 2^3 = 8$

1)

A \ BC	00	01	11	10
0	000	001	011	010
1	100	101	111	110

2)

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$A\bar{B}\bar{C}$	$A\bar{B}C$	$\bar{A}BC$	$\bar{A}B\bar{C}$
A	$AB\bar{C}$	$AB\bar{C}$	ABC	$AB\bar{C}$

3)

A \ B	$\bar{B}$	B
$\bar{A}$	0	1	3	2
A	4	5	7	6

4 Variable K- Map

$2^n = 2^4 = 16$

AB \ CD	00	01	11	10
00	0000 0	0001 1	0011 3	0010 2
01	0100 4	0101 5	0111 7	0110 6
11	1100 12	1101 13	1111 15	1110 14
10	1000 8	1001 9	1011 11	1010 10

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}CD$	$\bar{A}\bar{B}C\bar{D}$
$\bar{A}B$	$\bar{A}B\bar{C}\bar{D}$	$\bar{A}B\bar{C}D$	$\bar{A}BCD$	$\bar{A}BC\bar{D}$
AB	$AB\bar{C}\bar{D}$	$AB\bar{C}D$	$ABCD$	$ABC\bar{D}$
$A\bar{B}$	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}D$	$A\bar{B}CD$	$A\bar{B}C\bar{D}$

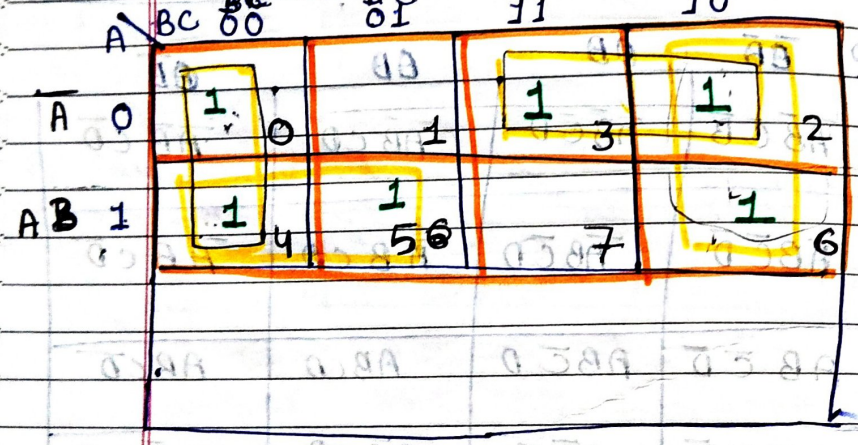
1) $f(A,B,C) = \Sigma m(0,1,2,3,4,5,6)$

* $\Sigma m()$ minterm → SOP when F = 1 so put 1 in k Map

* ΠM → Maxterm → POS → F = 0 so put '0' in k map *

Ans 3 veriable k map

$2^3 = 8$

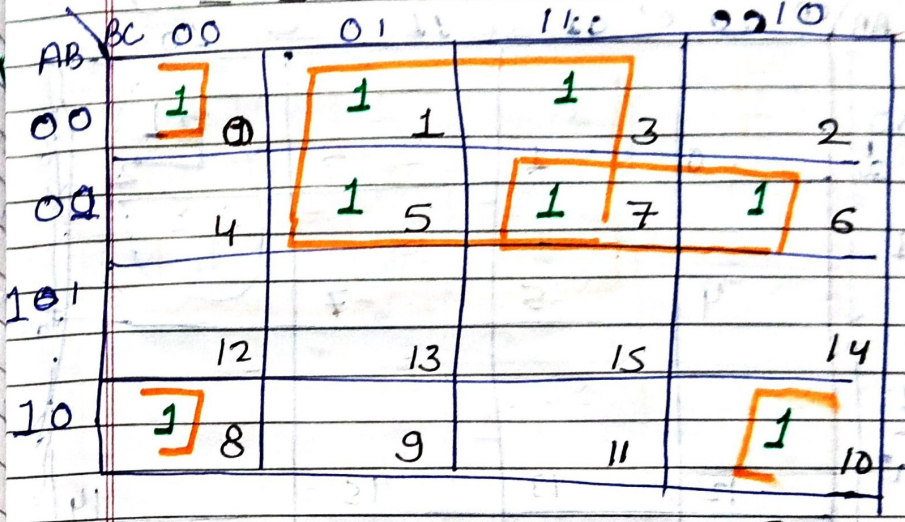


$f = \bar{C} + A\bar{B} + \bar{A}B$

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

1) $F = (a,b,c,d)$

$= \Sigma(0,1,3,5,6,7,8,10)$

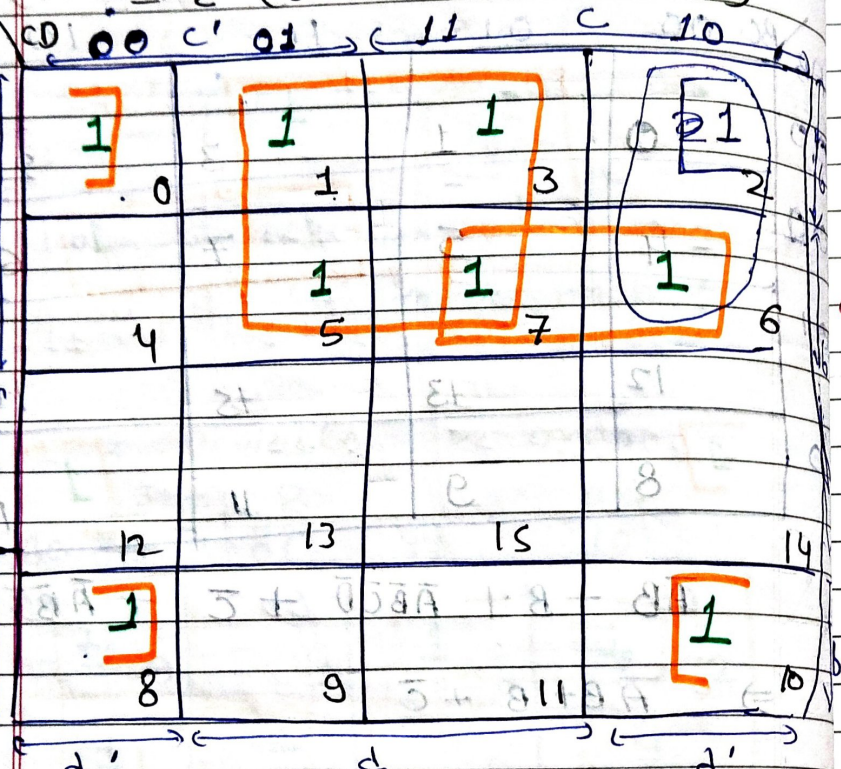


$\bar{A}B + B + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{C} + \bar{A}\bar{B}\bar{C}\bar{D}$

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

2> $F = (a, b, c, d)$

$= \Sigma\,(0, 1, 2, 3, 5, 6, 7, 8, 10)$



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

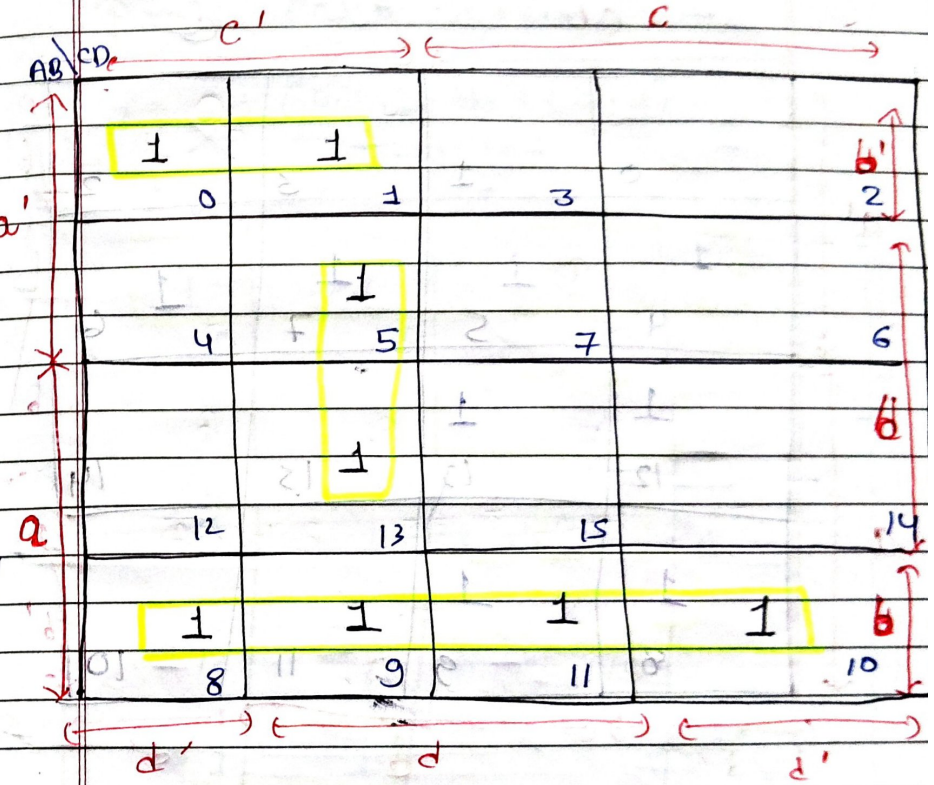
$\cancel{ad + b'd' + abc + a\bar{c}d'}$

$a'd + b'd' + a'bc + a'cd'$

H.W

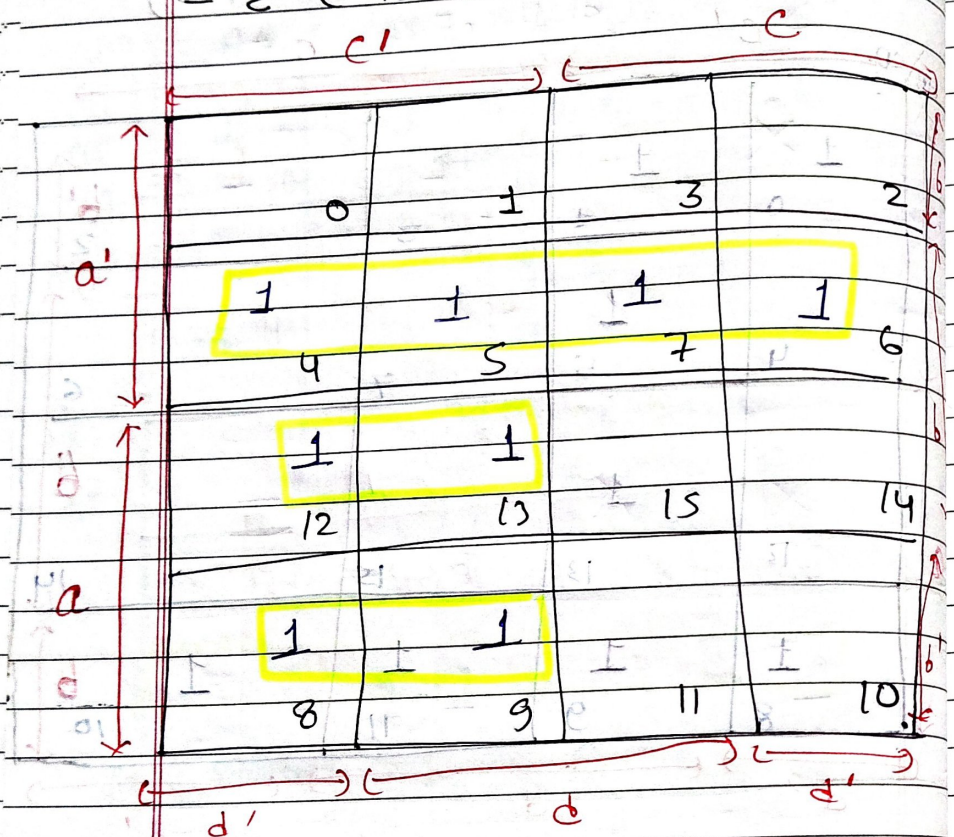
1> $F = (a, b, c, d)$

$= \Sigma\,(0, 1, 5, 8, 9, 10, 11, 13)$



$a'b'c' + bc'd + ab'$

2) $f(a,b,c,d) = \Sigma(4,5,6,7,8,9,12,13)$



$a'b + ab\,c' + ab'c'$

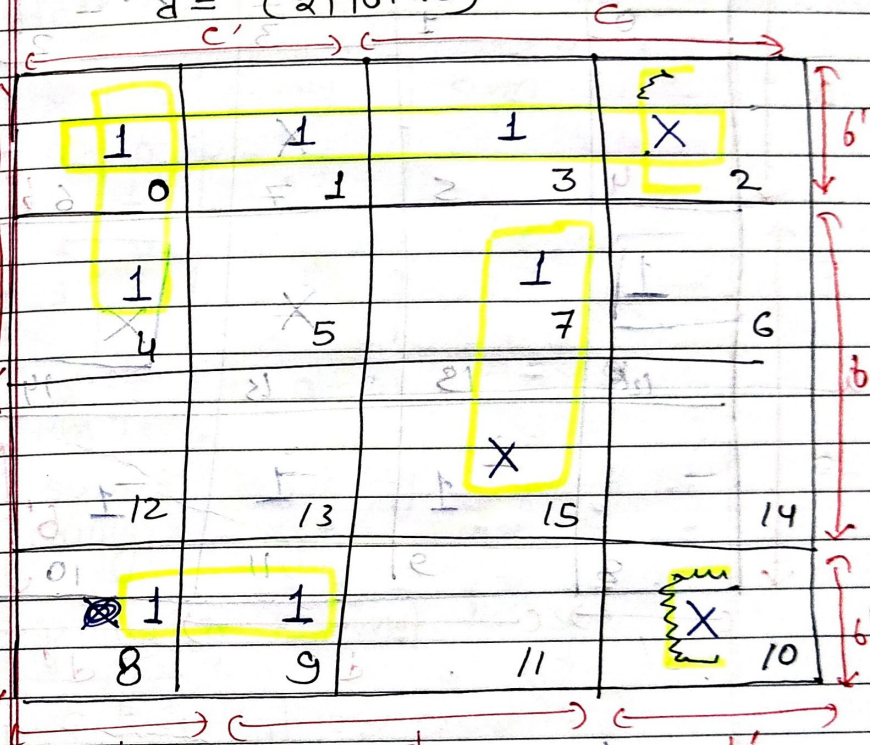
8/10/24

Digital Techniques

Don't Care :-

$f = \Sigma(0,1,3,4,7,8,9)$

$d = (2,10,15)$



$a'b' + a'c'd' + ab'c' + bcd + b'c'd'$

FREEMIND

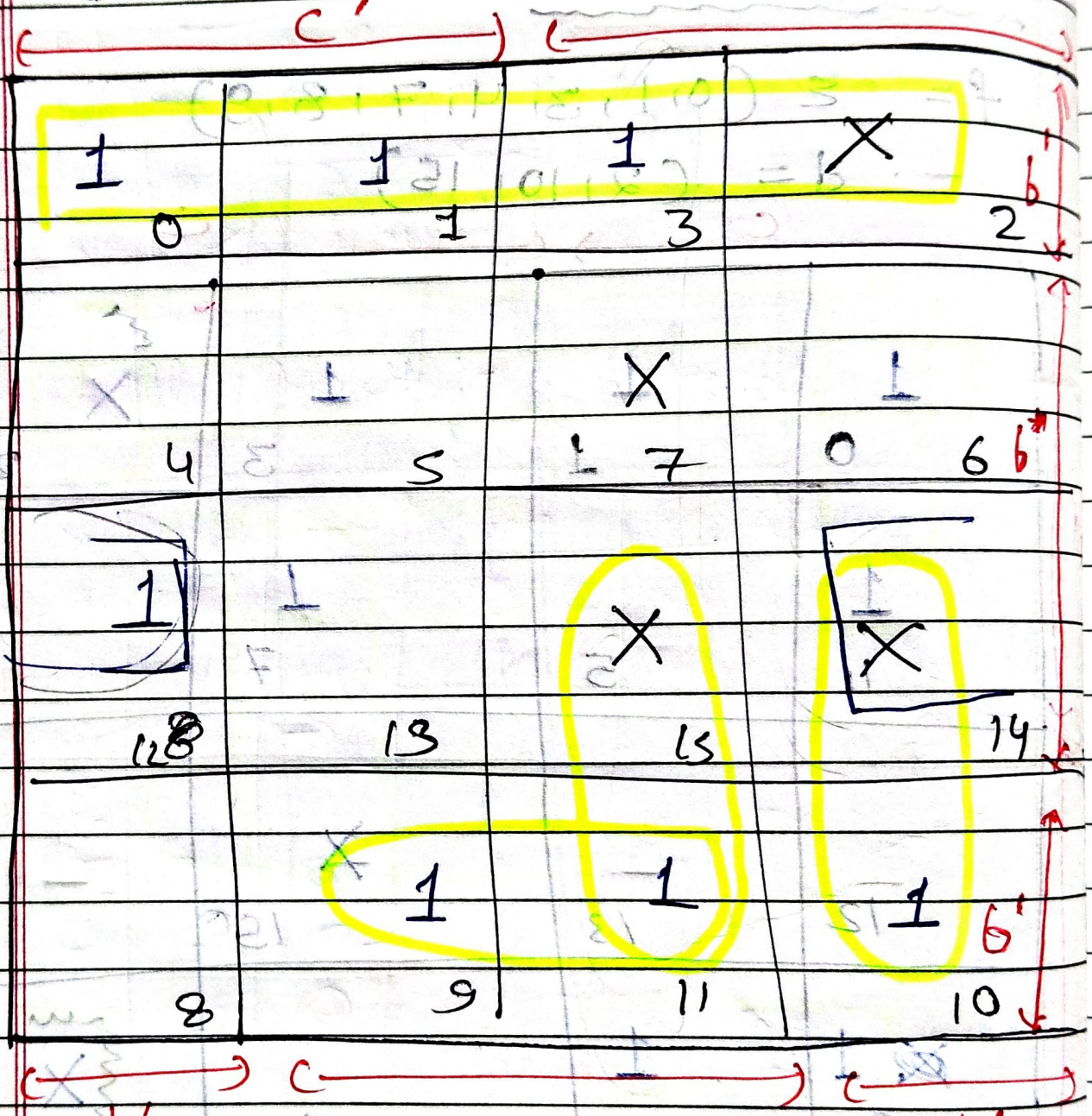
Date

Page

*

2) $f(a,b,c,d) = \Sigma(0,1,3,9,10,11,12)$

$d = (2,7,14,15)$



$a'b' + ab'd + acd + acd'$

FREEMIND

Date

Page

Combinational Logic Circuit

Half Order :-

Half Adder :- (2 I/P 2 O/P) => Adds 2 bits

		MSB bit	LSB Bit
x	y	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

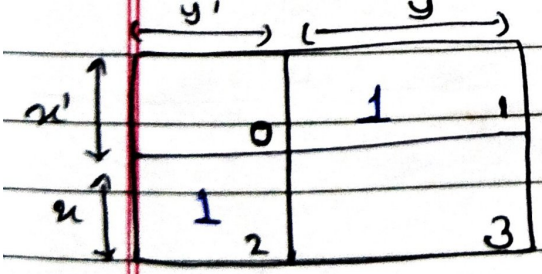
$2^2 = 4 = 0 - 3$

Min Term

$Sum = x'y + xy'$

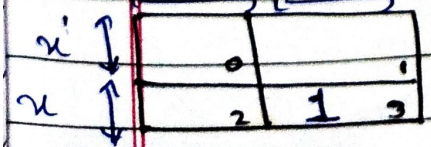
$Carry = xy$

Draw a k map for sum



$x'y + xy'$

y' draw y a k map for carry



$C = xy$

14/04/24

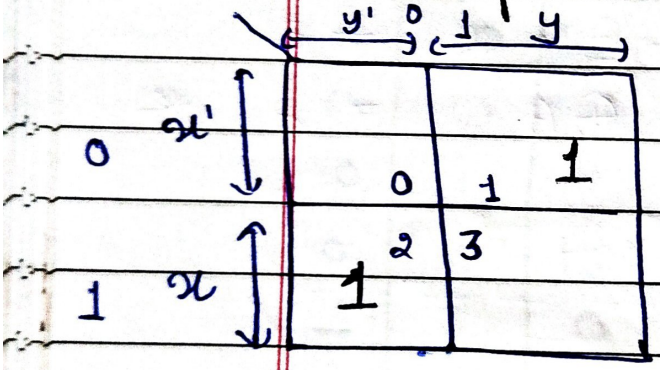
Half Subtractor

for 2 Variables $= 2^n = 2^2 = 4$

x - y

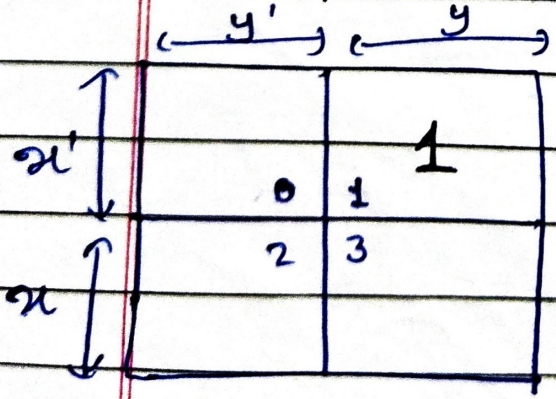
x	y	Subt	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-Map for subtraction



Sub = $x'y + xy'$

K-Map for Borrow



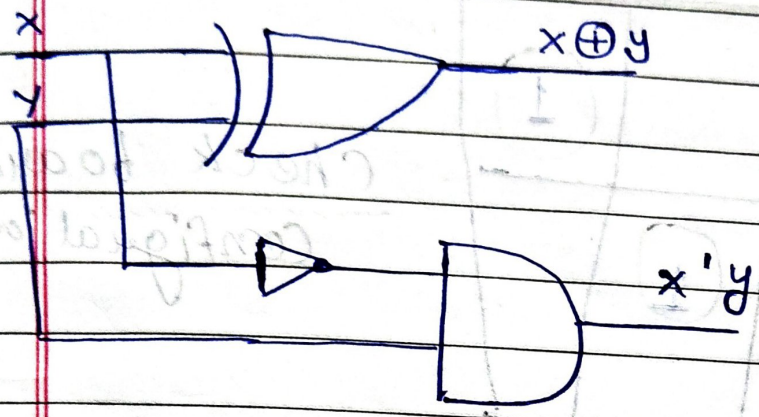
B = $x'y$

FREEMIND

Date

Page

Draw a gate for subtraction and Borrow



draw a invoter in x input

Full Subtractor :-

X	Y	Z	Sub	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$S = x'y'z + x'yz' + xy'z' + xyz$

$B = x'y'z + x'yz' + x'yz + xyz$

FREEMIND

Date

Page

—: Decoder :—

- It is a multiple input & multiple output.
- Decoder is a combinational circuit that converts n lines input into 2^n lines of output

- Application of decoders are converting binary code to other code
  - * Binary to octal
  - * " " to Hex
  - * " " to decimal
  - :

x	y	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

xy	
0 0	x'y'
0 1	x'y
1 0	xy'
1 1	xy

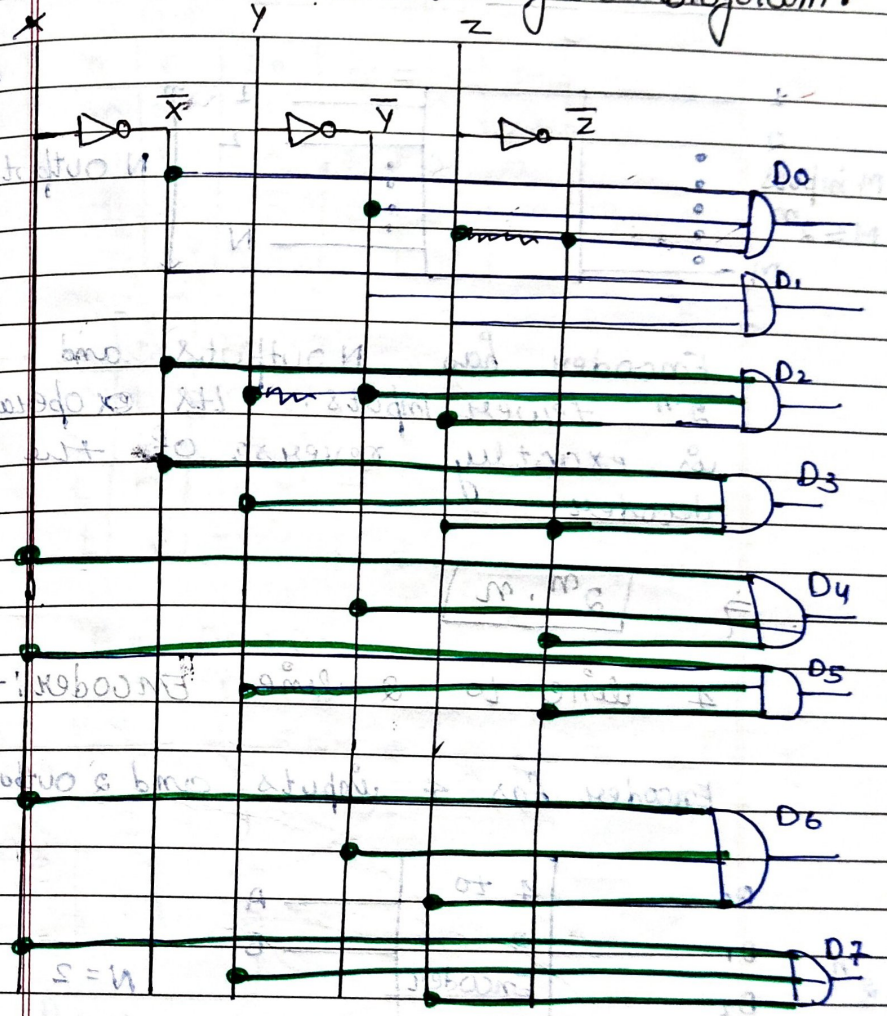
2) 3x8 Decoder (n inputs)
($M \leq 2^n$ outputs)

$n \cdot 2^n \Rightarrow 3 \cdot 2^3 = 3x8$

x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

	x	y	z	MinTerm
D_0	0	0	0	$x'y'z'$
D_1	0	0	1	$x'y'z$
D_2	0	1	0	$x'yz'$
D_3	0	1	1	$x'yz$
D_4	1	0	0	$xy'z'$
D_5	1	1	0	xyz'
D_6	1	0	1	$x*y'z$
D_7	1	1	1	xyz

3x8 line Decoder Logic Diagram :-



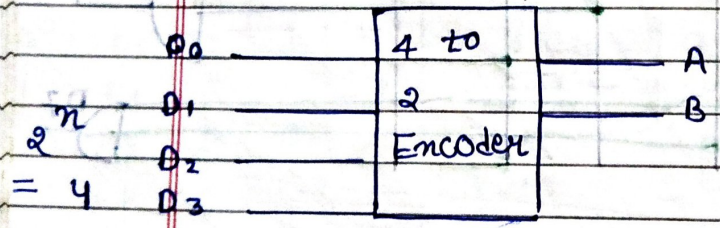
Encoder :-

Encoder has N outputs and 2^n fewer inputs. Its operation is exactly reverse of the decoder.

⇒ $\boxed{2^n \cdot n}$

4 line to 2 line Encoder :-

Encoder has 4 inputs and 2 outputs



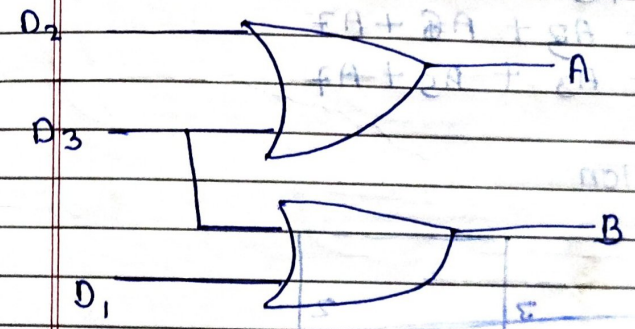
Truth Table

output		input			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

A	B	Eq
0	0	$x'y'$
0	1	$x'y$
1	0	xy'
1	1	xy

$A = D_2 + D_3$

$B = D_1 + D_3$

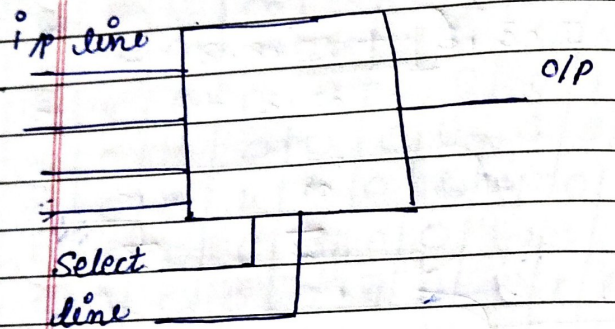


Date
Page

Multiplexer :-

MUX

Combinational CKT :-

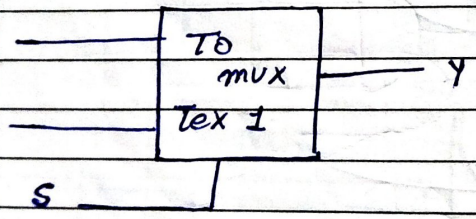


Select line = n

i/p line = 2^n

$m = 2^n$

o/p line = 1



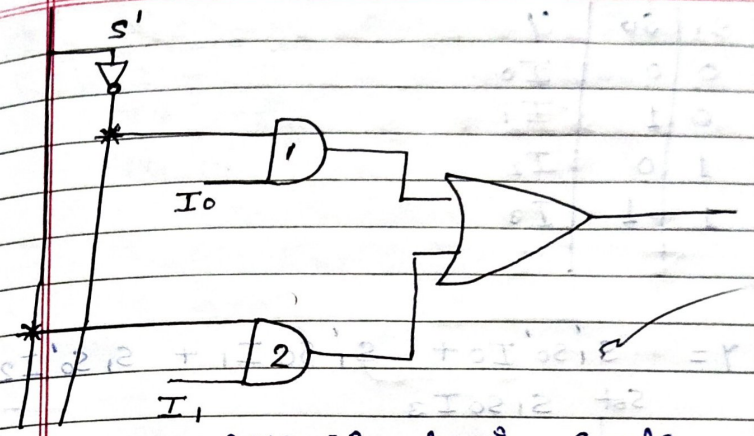
select line → 1

$2^1 = 2$

S	Y
0	I_0
1	I_1

$Y = S'I_0 + SI_1$

5



AND OR logic Gate

2) find 4X1 MUX

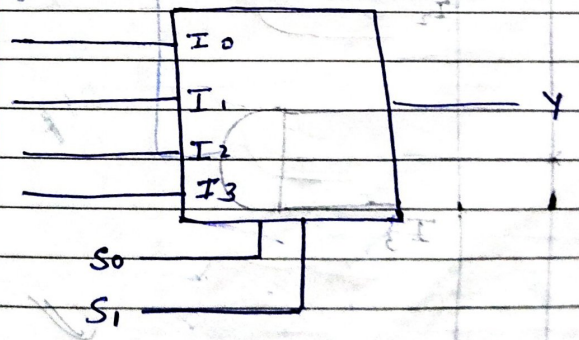
i/p = 4

i/p line = $2^2 = 4$

select line = 2

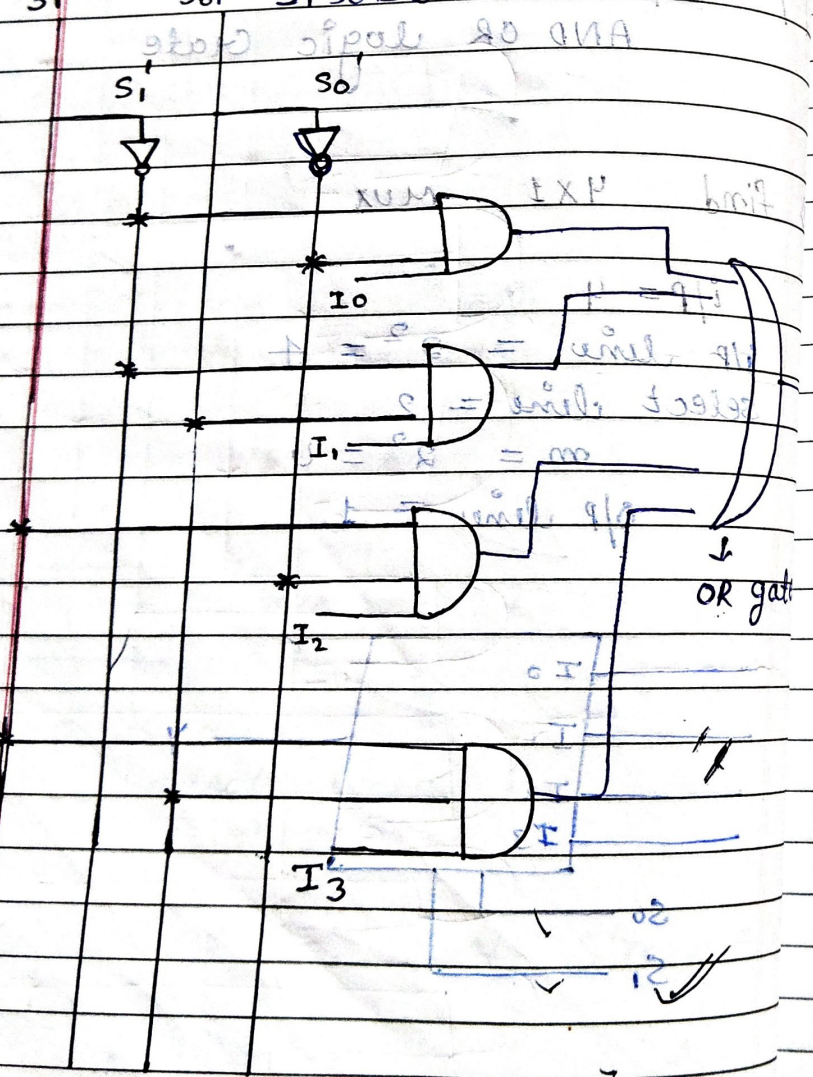
$m = 2^2 = 4$

o/p line = 1



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

$Y = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$



H.W

3) 8 by 1 MUX

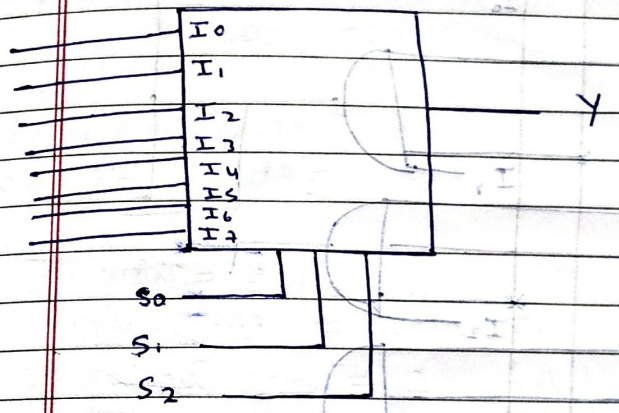
i/p = 8

i/p line = $2^n = 2^3 = 8$

$n = 3$

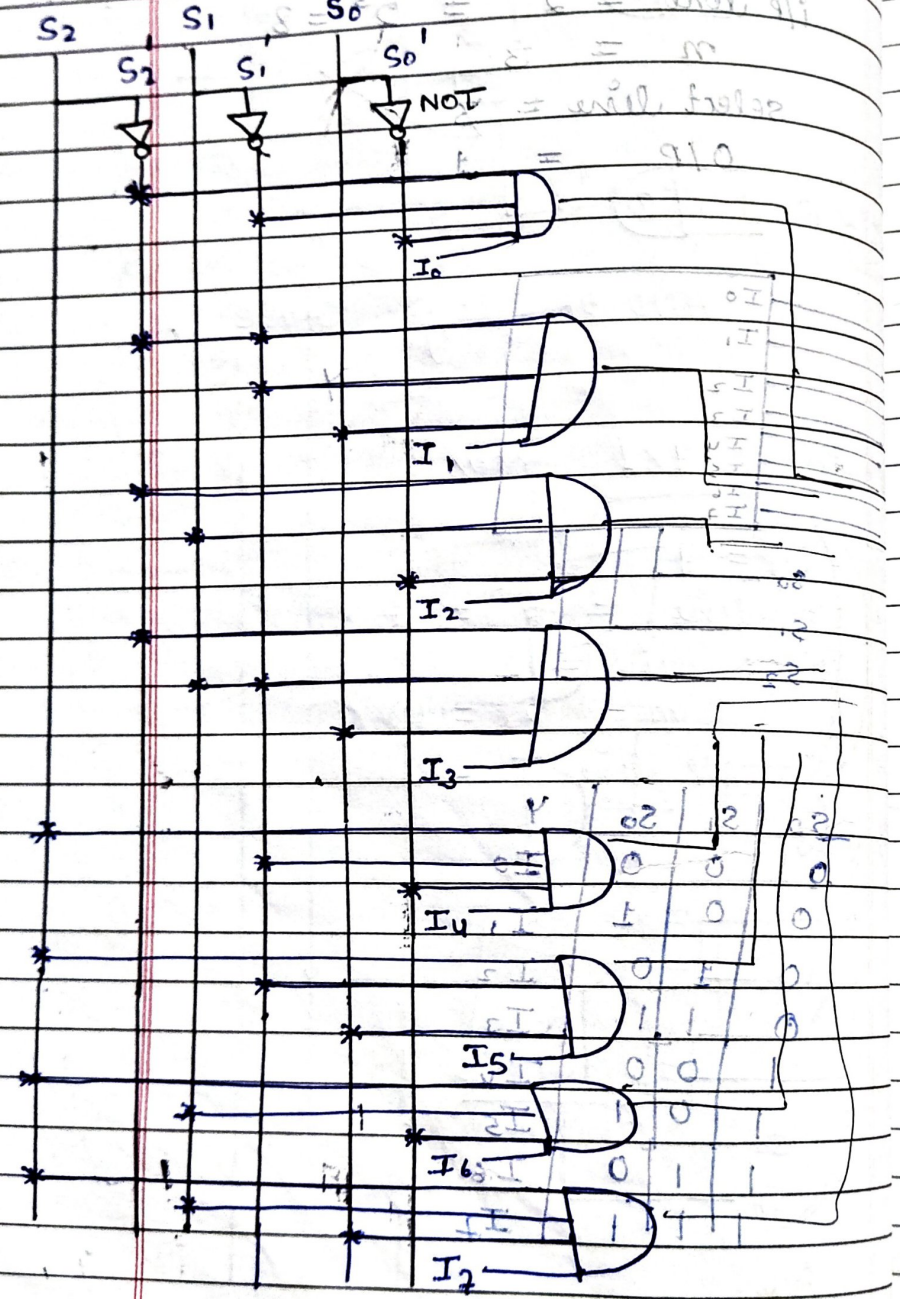
select line = 3

O/P = 1



S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

$S_2' S_1 S_0' I_0 + S_2' S_1 S_0 I_1 + S_2 S_1' S_0' I_4 + S_2 S_1' S_0 I_5 + S_2 S_1 S_0' I_6 + S_2 S_1 S_0 I_7$



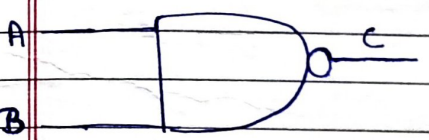
UNIVERSAL LOGIC

Universal logic gates are specific types of logic gate that can be used to implement any Boolean function without the need for additional gate type.

The two Primary universal logic gate are:

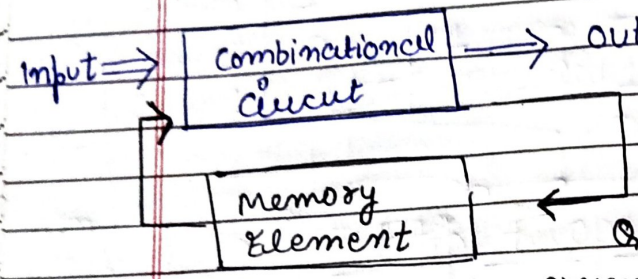
1) NAND Gate
2) NOR Gate

1) NAND Gate



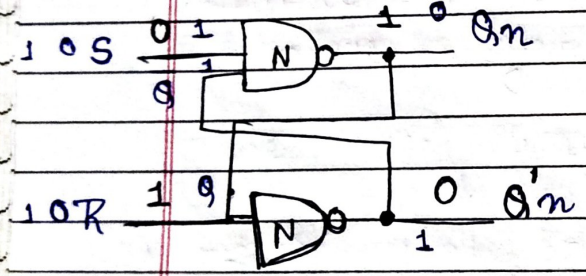
A B	C
0 0	1
0 1	
1 0	
1 1	

Sequential Circuits :-



Sequential circuits are digital circuits whose output depends not only the present input but also on the past history.

Latch :- SR Latch using NAND Gate



S	R	output Qn+1
0	0	1
0	1	1
1	0	1
1	1	0

one of the input zero ⇒ output-1

$\overline{1 \cdot Q} = 0 + Q$

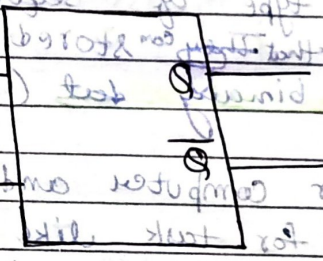
$\overline{1} \cdot \overline{Q} = 0 + Q$

$= \overline{Q}$

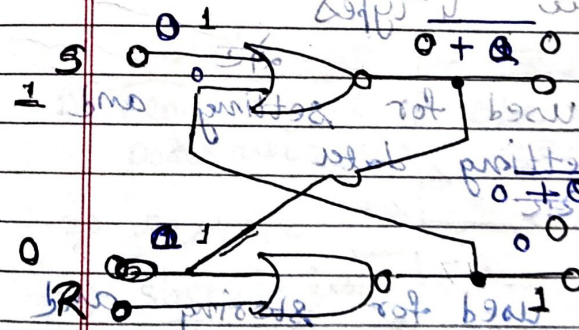
S	R	Q(n+1) (Next State)
0	0	invalid state
0	1	1 Reset
1	0	0 Set
1	1	Hold state ⇒ previous value ही आती है,

SR Latch [Q / $\overline{Q}$] same output

An SR latch is fundamental digital circuit that can store a single bit of information.



SR Latch using NOR GATE



S	R	Q(n+1)
0	0	Q
0	1	0
1	0	1
1	1	0

one of the input one output is zero

S	R	Q(n+1)
0	0	Hold state
→ 0	1	0
→ 1	0	1
1	1	Invalid stage

Flip flop

A flip-flop is basic memory element in digital electronic it is a type of sequential circuit. ~~that they can~~ stored a single bit of binary data (0 or 1)

used for computer and digital system for task like data storage, synchronization, Count

These are 4 types

1) SR ⇒ used for setting and resetting data

2) D ⇒ used for storing and transfaring data

3) J-K flip-flop ⇒ advance version of the SR flip flop includes toggle mode

4) T - flip-flop ⇒ used for toggling data between 0 and 1

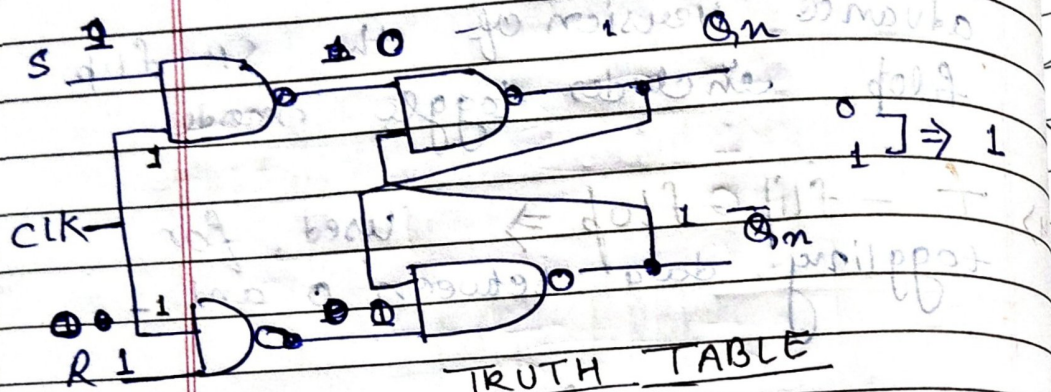
flip-flop ~~का~~ clock signal ~~द्वारा~~ used for making

1) circuits
2) Register
3) counter
4) memory module

Difference

Latch	flip-flop
1) clock signal Does not require	operates based on clock signal
2) faster	slower
3) simpler design	more complex
4) used in asynchronous circuit	used in synchronous circuit

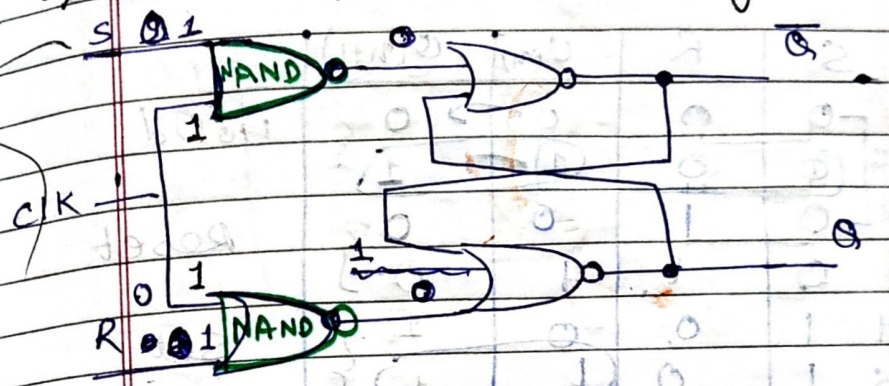
1) SR flip-flop ⇒ Using NAND



TRUTH TABLE

CLOCK	S	R	Qn+1
0 – Not trigged	X	X	Qn
1 – Trigger	0	0	Hold
Trigger	0	1	0 – Reset
Trigger	1	0	1 Set
Trigger	1	1	Invalid

2) SR - flip flop ⇒ Using NOR GATE



TRUTH TABLE

CLK	S	R	Qn+1
X Not trigg	X	X	Qn
1 Trig	0	0	Hold
1 Trig	0	1 →	0
1 Trig	1	0	1
	1	1	Invalid

NAND or NOR ही flip-flop Truth Table, are same

$$Q_{n+1} = \bar{R}Q + S$$

$2^3 = 8$

Characteristics Table of SR

	S	R	Q(n)	Q(n+1)	
0	0	0	0	0	Hold
1	0	0	1	1	
2	0	1	0	0	Reset
3	0	1	1	0	
4	1	0	0	1	Set
5	1	0	1	1	
6	1	1	0	X	condition invalid
7	1	1	1	X	

SR	Qn		Qn+1	
00	0	0	0	Hold
01	0	1	0	Reset
10	1	0	1	Set
11	1	1	X	Invalid

Equation

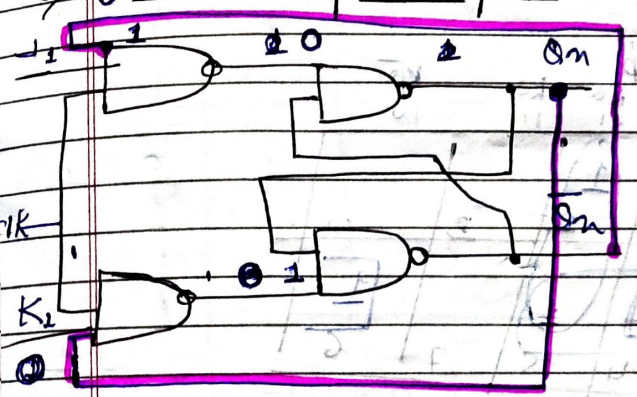
	RQ	R̄Q	RQ̄	R̄Q̄
S̄	0	1	3	2
S	1	5	7	6

$$S + \bar{R}Q = Q_{n+1}$$

Exitation Table

Qn	Q(n+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

2) J-K flip-flop



SR	Qn+1	
00	Hold	
01	0	
10	1	
11	invalid	solve this problem → Togal → Bari

Case 1: Qn = 1, Qn+1 = 0

Case 2: Qn = 0, Qn+1 = 1

→ Togle

S	R	K	Qn+1	
0	0	0	Hold	
0	X	1	0	Reset
1	0	X	1	set
1	1	X	Q̄n	Togle

Characteristics Table

	J	K	Q(n)	Q(n+1)	
	0	0	0	0	Hold
1	0	0	1	1	
2	0	1	0	0	Reset
3	0	1	1	0	
4	1	0	0	1	set
5	1	0	1	1	
6	1	1	0	1	
	1	1	1	0	

K-Map

Q \ JQ̄	JQ̄ 00	JQ̄ 01	JQ 11	JQ 10
Q	0	1	3	2
Q̄	1	1	5	1 (6)
	4		7	

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

Excitation Table

Qn	Q(n+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

3) D flip-flop

Storage device

input → output

D	Qn+1
0	0
1	1

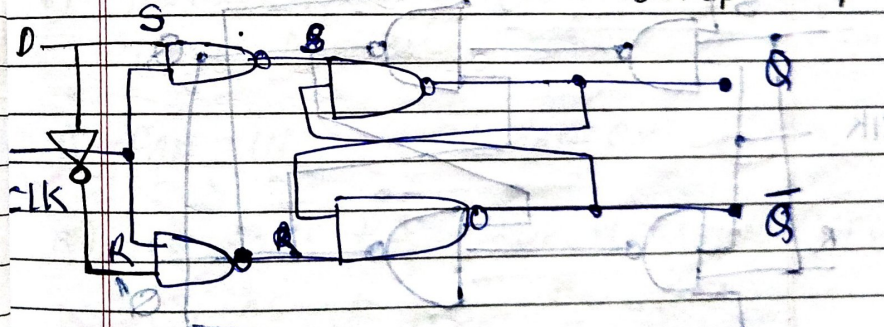
Truth Table

Characteristics Table

D	Qn	Qn+1
0	0	0
0	1	0
1	0	1
1	1	1

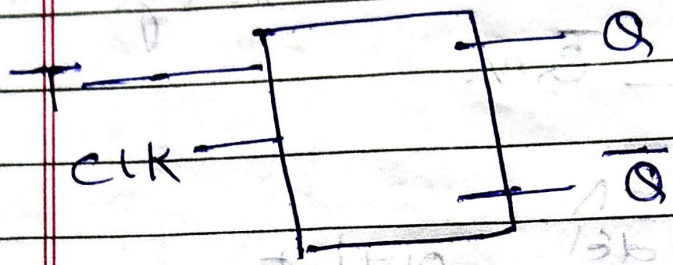
$$Q_{n+1} = D$$

D-flip-flop



Date
Page

4> T- flip- flop



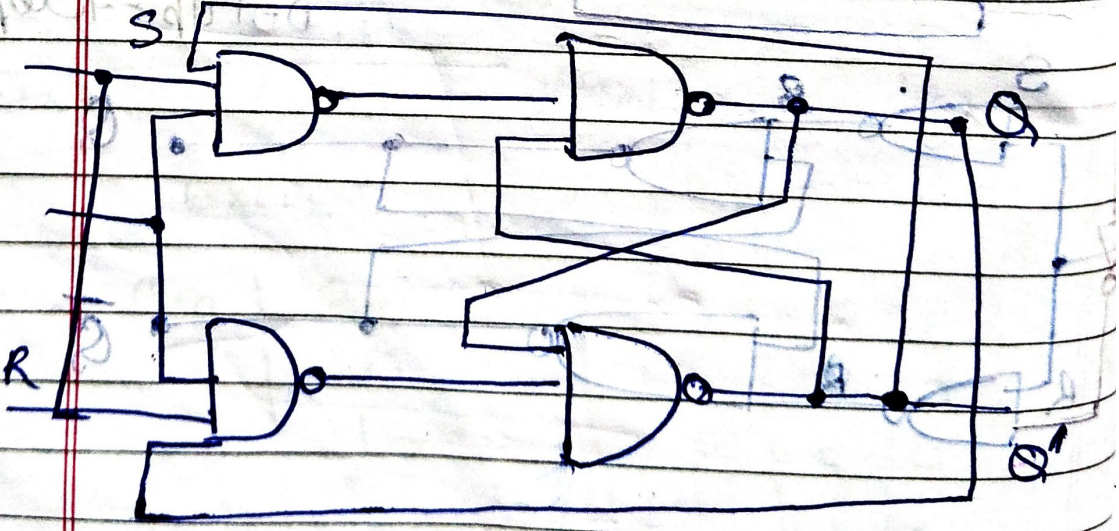
T	Q_{n+1}
0	Q_n
1	$\overline{Q_n}$

Characteristic Table

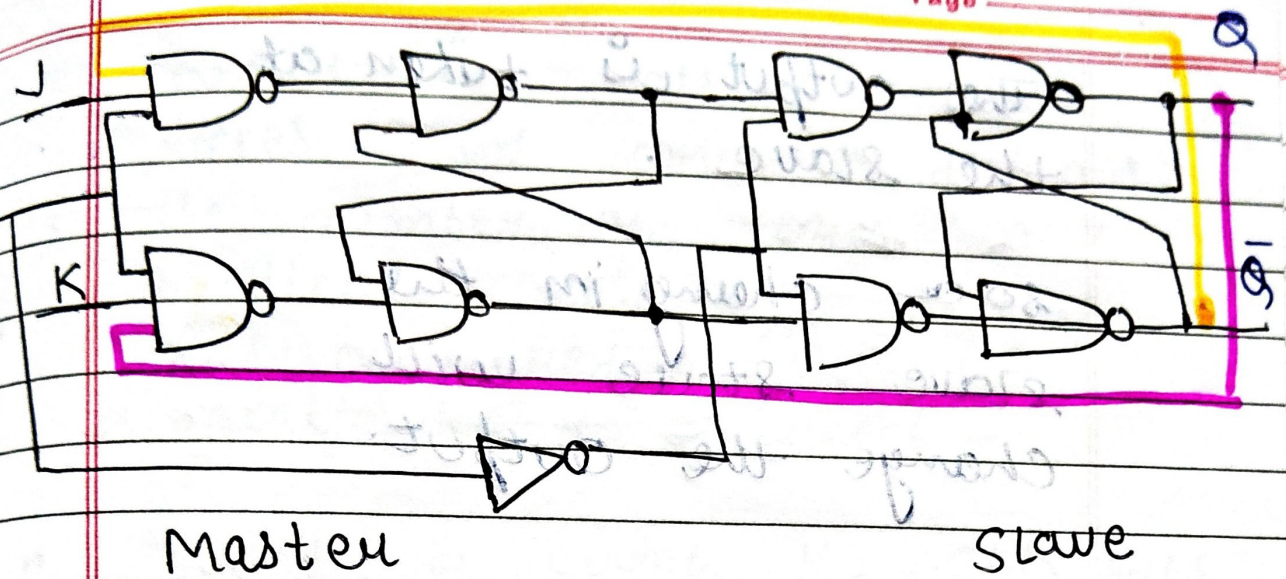
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

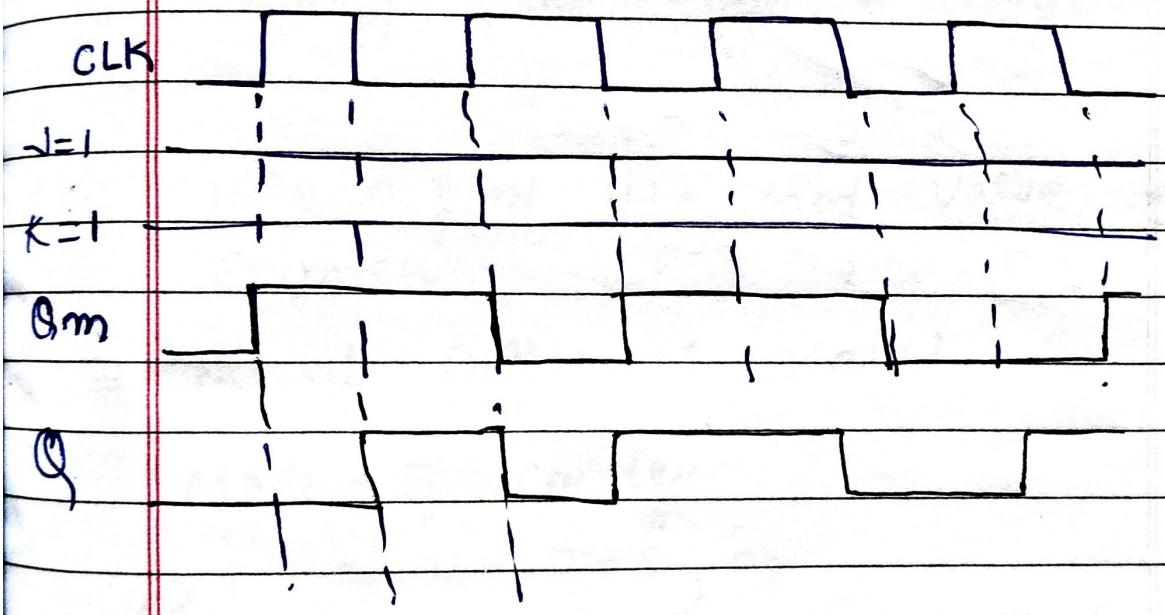
Toggle

$$Q_{n+1} = T \oplus Q_n$$



Master slave





A Master slave flip-flop is a digital circuit

↓

That stores binary information

↓

and is made up of two gated flip-flop master and slave

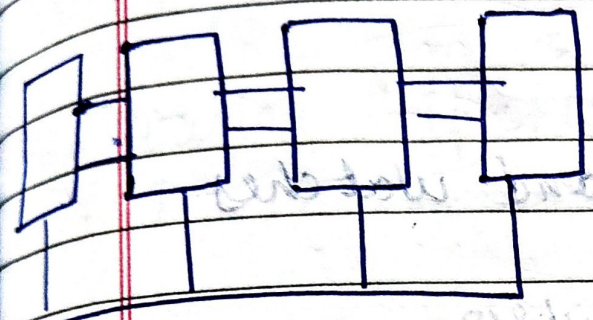
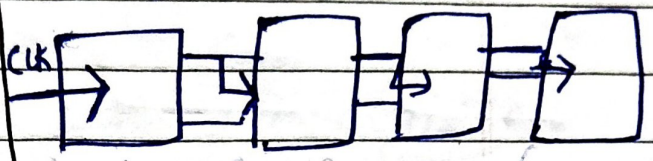
Master ⇒ Receives input ~~data~~ during the active edge of a clock signal

Slave ⇒ latches the data ~~activ~~ opposite during the signal

FREEMIND

Date

Page

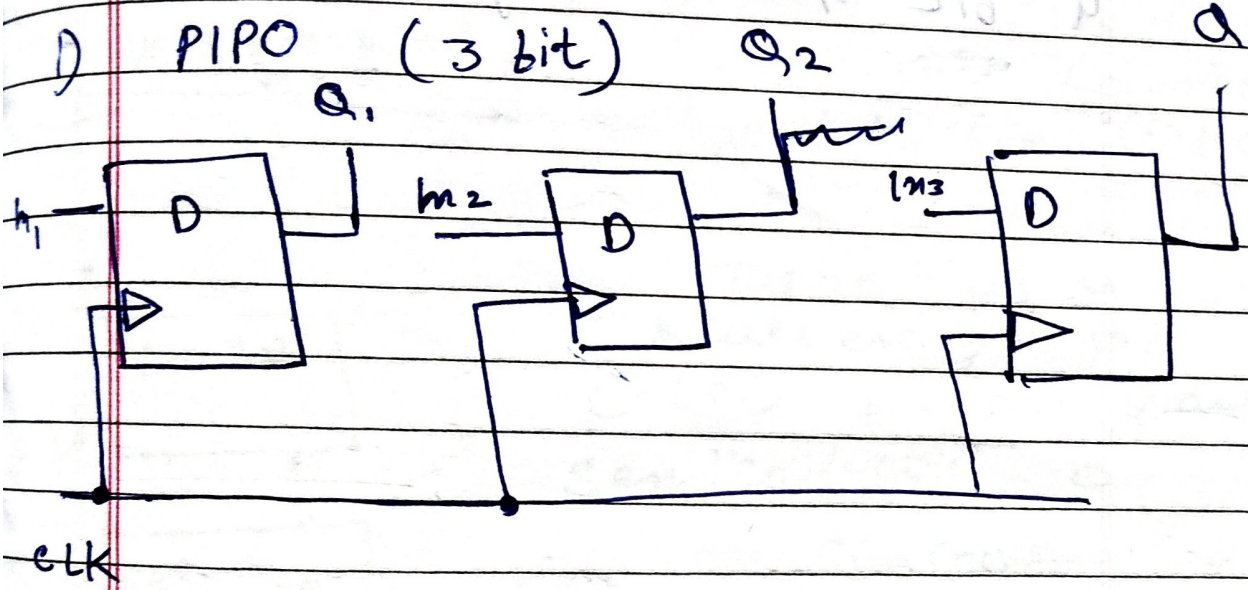
Synchronous	Asynchronous
	Ripple counter 
faster in operation	slower in operation
Complex Design	Easy
High cost	Low cost
Delay ~~24~~	Deady with
Propagation delay is very low	Propagation delay is higher
also know as Parallel counter	know as serial counter
Example : Ring counter, Johnson	Ripple Up Ripple down

Shift Register

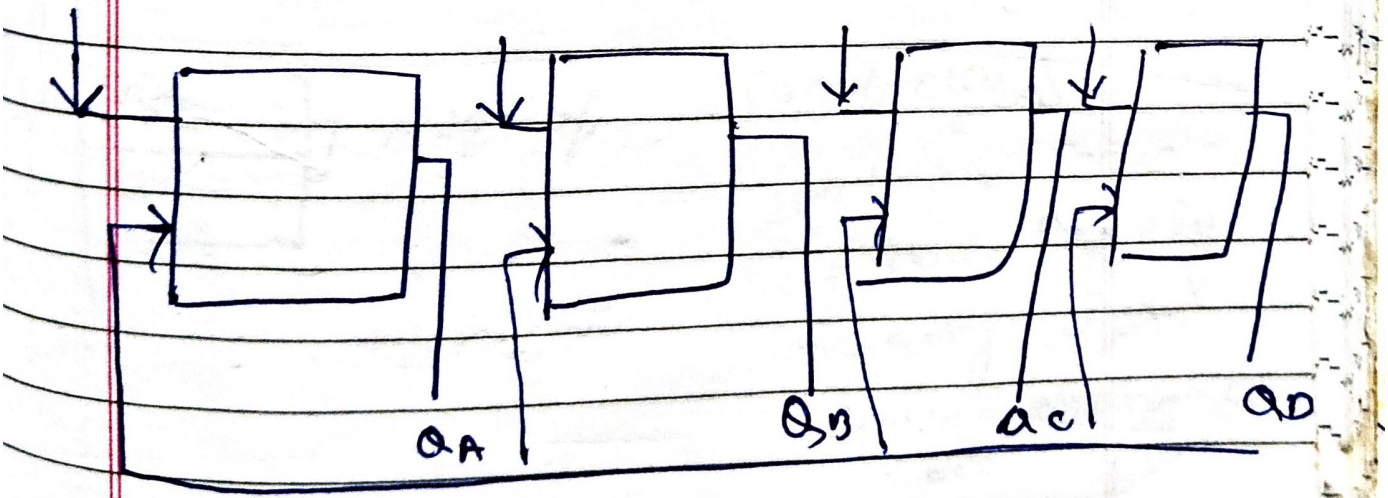
Date
Page

- shift Register are used to implement arithmetic operation.
- Basic flip-flop used in register D ff

1) PIPO (3 bit)



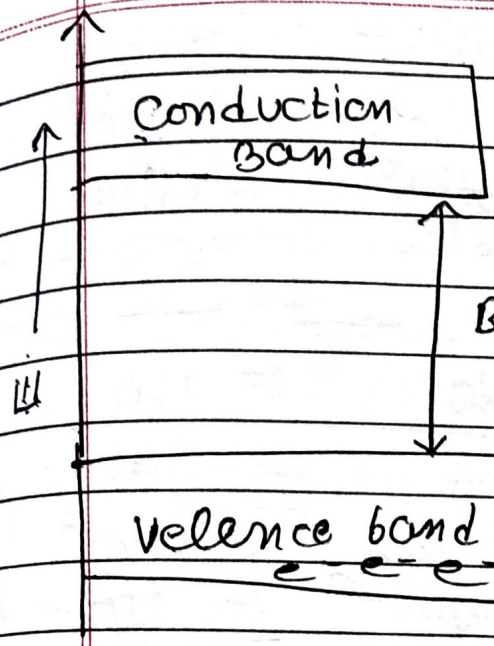
2) PIPO 4 bit



FREEMIND

Date

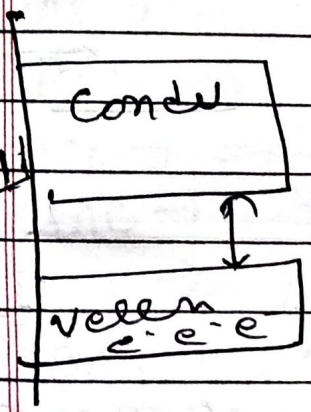
Page



large energy gap (>3eV) between V and C

Insulator

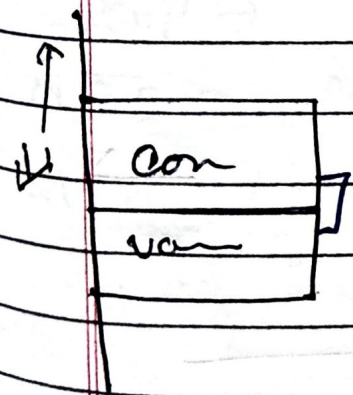
Electron cannot jump to the Conducti band so no current flow



There is a small energy gap (~1eV) between V and C

Semiconductor

Si, Ge



Conductors

electron can flow freely making it a good conductor of electri