

NUMBER SYSTEM

$(N)_R$ where $R = 10 \rightarrow$ Decimal

$8 \rightarrow$ Octal

$2 \rightarrow$ Binary

$16 \rightarrow$ HexaDecimal

Q- $10^0 = 1$. Prove.

$$\text{A-} \quad \text{let } x^0 = x^{n-n}$$

$$= x^n \times x^{-n}$$

$$= \cancel{x^n} \times 1 = \cancel{x^n} = 1$$

* Bits \rightarrow BINARY DIGITS

Binary \rightarrow BI + NARY

(2) + NUMBER SYSTEM

Dec

Bin

0

0

1

1

2

10

3

11

Q- $(143)_{10} \rightarrow (?)_2$

2 | 143

2 | 71 1

2 | 35 1

2 | 17 1

2 | 8 1

2 | 4 0

2 | 2 0

1

$(1001111)_2$

Q- $(10001111)_{10} \rightarrow (?)_2$

$$\begin{array}{r}
 | \quad | \quad | \\
 2^0 \times 1 = 1 \\
 2^1 \times 1 = 02 \\
 2^2 \times 1 = 04 \\
 2^3 \times 1 = 08 \\
 2^4 \times 1 = 128 \\
 \hline
 (143)_{10}
 \end{array}$$

Q- $(0.25)_{10} \rightarrow (?)_2$

$$\begin{array}{r}
 0.25 \times 2 = 0.5 \rightarrow 0 \\
 0.5 \times 2 = 1 \rightarrow 1
 \end{array}$$

Q- $(0.01)_2$

$$\begin{array}{r}
 2^{-1} \times 0 + 2^{-2} \times 1 \\
 = 2^{-2} = \frac{1}{4} = (0.25)_{10}
 \end{array}$$

Q- $(0.33)_{10} = (?)_2$

$$\begin{array}{r}
 0.33 \times 2 = 0.66 \rightarrow 0 \\
 0.66 \times 2 = 1.32 \rightarrow 1 \\
 1.32 \times 2 = 0.64 \rightarrow 0 \\
 0.64 \times 2 = 1.28 \rightarrow 1 \\
 1.28 \times 2 = 0.56 \rightarrow 0 \\
 0.56 \times 2 = 1.12 \rightarrow 1
 \end{array}$$

$$(0.010101)_2$$

Octal Number System

→ In case of octal no. system each digit of octal is represented 3-bit form

<u>Dec</u>	<u>Oct</u>	<u>Bin</u>
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111

Q- $(8)_{10} \rightarrow ? ()_8$

$$\begin{array}{r} 8 | 8 \\ \quad 1 \quad 0 \end{array} \qquad (10)_8$$

Q- $(10)_8 \rightarrow (?)_8$

$$\begin{array}{r} 1 \quad 0 \\ (001 \ 000)_2 \end{array}$$

Q- $(143)_{10} \rightarrow (?)_8$

$$\begin{array}{r} 8 | 143 \\ 8 | 17 \quad 7 \\ \quad 2 \quad 1 \end{array} \qquad \begin{array}{r} (217)_8 \\ \downarrow \\ (010\ 001\ 111)_2 \end{array}$$

Q: $(0.25)_{10} \rightarrow (?)_8$ [Ans. 0.25]

$$0.25 \times 8 = 2.00 \rightarrow 2$$

$$(0.2)_8$$

Q: $(0.2)_8 \rightarrow (?)_{10}$

$$8^{-1} \times 2 = \frac{2}{8} = \frac{1}{4} = (0.25)_{10}$$

~~30/08/24~~

Hexa Decimal Number System

In hexa decimal no. system each digit of hexa will be represented in 4-bit form

Dec	Hexa	Bin
-----	------	-----

0	0	0000
---	---	------

1	1	0001
---	---	------

2	2	0010
---	---	------

3	3	0011
---	---	------

4	4	0100
---	---	------

5	5	0101
---	---	------

6	6	0110
---	---	------

7	7	0111
---	---	------

8	8	1000
---	---	------

9	9	1001
---	---	------

10	A	1010
----	---	------

11	B	1011
----	---	------

12	C	1100
----	---	------

13	D	1101
----	---	------

14

E

1110

16116

15

F

1111

10

16

10

 $(100010000)_2$

$$\text{Q- } (239)_{10} \rightarrow (?)_{16}$$

16 [239]

14 15

 $(FF)_{16}$

↓

 $16' \times F = 15$ $(11101111)_2$

15

$$\text{Q- } (0.25)_{10} \rightarrow (?)_{16}$$

 $(239)_{10}$

$$0.25 \times 16 = 400 \downarrow$$

 $(0.4)_{16}$

$$= 4/16 = 1/4$$

 $(0.25)_{10}$

Q- Find The radix of The no.s for a given quadratic eqⁿ

$$x^2 - 10x + 31 = 0$$

$$\text{is } x = 5 \& 8$$

A-

$$x = 5, 25 - 50 + 31$$

$$\alpha + \beta = (10)_R \quad \text{Let radix} = R$$

$$\alpha \beta = (31)_R \quad (2)$$

$$\text{Case 1 } 5 + 8 = R' \times 0 + R' \times 1$$

$$\Rightarrow R = 13$$

$$\text{Case } ②, 5 \times 8 = R^o \times 1 + R' \times 3$$

$$\Rightarrow 3R = 39$$

$$\Rightarrow \boxed{R = 13}$$

$$\begin{aligned}x &= 5 & x^2 - 10x + 31 \\&= 5^2 - (13^o \times 0 + 13' \times 1)5 + 13^o \times 1 + 13' \times 3 \\&= 25 - 65 + 1 + 39 \\&= 0\end{aligned}$$

$$\begin{aligned}x &= 8 & x^2 - 10x + 31 \\&= 8^2 - (13^o \times 0 + 13' \times 1)8 + 13^o \times 1 + 13' \times 3 \\&= 64 - 104 + 1 + 39 \\&= 0\end{aligned}$$

~~H.W.~~

$$(271.25)_{10} = (?)_3 = (?)_5 = (?)_7 = (?)_{12}$$

i) $3 | 271 \quad (101001)_3$

$$3 | 90 \quad 01$$

$$3 | 30 \quad 0 \quad 0.25 \times 3 = 0.75 \rightarrow 0$$

$$3 | 10 \quad 0 \quad 0.75 \times 3 = 2.25 \rightarrow 2$$

$$3 | 3 \quad 1 \quad 0.25 \times 3 = 0.75 \rightarrow 0$$

$$1 \quad 0 \quad 0.75 \times 3 = 2.25 \rightarrow 2$$

$$\therefore (271.25)_{10} \rightarrow (101001.0202)_3$$

ii) $5 | 271 \quad (2041)_5$

$$5 | 54 \quad 1$$

$$5 | 10 \quad 4 \quad 0.25 \times 5 = 1.25 \rightarrow 1$$

$$2 \quad 0 \quad 0.25 \times 5 = 1.25 \rightarrow 1$$

$$0.25 \times 5 = 1.25 \rightarrow 1$$

$$\therefore (271.25)_{10} \rightarrow (2041.1111)_5$$

* Binary Addition

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$1 + 1 = 0$ with carry 1

$$\begin{array}{r}
 100 \\
 + 111 \\
 \hline
 1011
 \end{array}
 \quad
 \begin{array}{r}
 1011 \\
 + 1101 \\
 \hline
 11000
 \end{array}$$

* Binary Subtraction

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$0 - 1 = 1$ with borrow 1

$$\begin{array}{r}
 10100 \\
 - 111 \\
 \hline
 001
 \end{array}
 \quad
 \begin{array}{r}
 11101 \\
 - 01011 \\
 \hline
 1000
 \end{array}$$

$$\begin{array}{r}
 1000 \\
 - 111 \\
 \hline
 01
 \end{array}
 \quad
 \begin{array}{r}
 \rightarrow 4 \\
 - 7 \\
 \hline
 -3
 \end{array}
 \quad
 \text{* Fails for negative nos.}$$

* 1's & 2's Complement

$$\begin{array}{r}
 1's \text{ of } 100 \rightarrow 100 \\
 \hline
 011
 \end{array}$$

$$\begin{array}{r} 1000 \\ 1010 \end{array} \xrightarrow{1's} \begin{array}{r} 0111 \\ 0101 \end{array}$$

$$\begin{array}{r} 100 \\ -111 \end{array} \quad \begin{array}{r} 111 \\ +100 \end{array} \xrightarrow{1's} \begin{array}{r} 000 \\ 100 \end{array}$$

* If input bit = output bit
 \Rightarrow -ve i.e. The output is The
 100 $1's$ of actual answer.
 $\downarrow 1's$
 $\underline{011}$

$$\begin{array}{r} 111 \\ 100 \\ ? \end{array} \quad \begin{array}{r} 100 \\ +111 \end{array} \xrightarrow{1's} \begin{array}{r} 011 \\ 1010 \end{array}$$

* If input bit \neq output bit
 \Rightarrow +ve The extra bit will be
 added to no.

$$\text{MSB} \leftarrow \begin{array}{r} 1010 \rightarrow \text{LSB} \\ \cancel{\times} \cancel{1} \cancel{1} \\ \underline{011} \end{array}$$

$$2's \rightarrow 1's + 1$$

$$100 \xrightarrow{1's} 011$$

$$\underline{100}$$

$$\begin{array}{r}
 100 \\
 - 111 \\
 \hline
 001 \\
 + 100 \\
 \hline
 101 \quad \text{Eve} \\
 \downarrow \\
 010 + 1 = \boxed{011}
 \end{array}$$

$$\begin{array}{r}
 111 \\
 - 100 \\
 \hline
 100 \\
 + 111 \\
 \hline
 101 \quad \text{(+ve)}
 \end{array}$$

* For +ve, The extra bit will be cancelled. in 2's complement.

* 9's & 10's

→ Applicable for decimal no-system

$$\begin{array}{r}
 82 \xrightarrow{9's} 99 \\
 - 82 \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 13 \xrightarrow{9's} 17 \\
 - 13 \\
 \hline
 86
 \end{array}$$

$$\begin{array}{r}
 43 \quad 69 \xrightarrow{9's} 30 \\
 - 69 \quad + 43 \\
 \hline
 73 \quad \text{(-ve)} \\
 \downarrow 9's \\
 126
 \end{array}$$

$$\begin{array}{r}
 69 \\
 -43 \\
 \hline
 26
 \end{array}
 \quad
 \begin{array}{r}
 43 \xrightarrow{9^s} 56 \\
 +69 \\
 \hline
 125 \rightarrow (+ve) \\
 \curvearrowleft 7+1
 \end{array}$$

$$\underline{10^s} \rightarrow 9^s + 1$$

$$\begin{array}{r}
 43 \\
 -71 \\
 \hline
 29
 \end{array}
 \quad
 \begin{array}{r}
 71 \xrightarrow{9^s} 28 \\
 +1 \\
 \hline
 72 \rightarrow (-ve)
 \end{array}$$

$$71 \rightarrow 43 \rightarrow 56$$

$$-43 \rightarrow 71 \rightarrow 57$$

$$\begin{array}{r}
 +71 \\
 \hline
 128 \rightarrow (+ve)
 \end{array}$$

iii) $7 | \underline{271}$ $(535)_7$

$$7 | \underline{385}$$

$$5 \quad 3 \quad 0.25 \times 7 = 1.75 \rightarrow 1$$

$$0.75 \times 7 = 5.25 \rightarrow 5$$

$$0.25 \times 7 = 1.75 \rightarrow 1$$

$$\therefore (271.25)_{10} \rightarrow (535.151)_7$$

iv) $12 \overline{) 271}$ $(IA7)_{12}$

$$\begin{array}{r} 12 \overline{) 22} \\ 12 \quad 7 \\ \hline 10(A) \\ 0.25 \times 12 = 3.0 \rightarrow 3 \end{array}$$

$$\therefore (271.25)_{12} \rightarrow (IA7.3)_{12}$$

~~02/09/24~~ ~~31 Dec 2023 Tuesday~~
 Non-terminating fraction without a
Floating Point Arithmetic

+	→ sign				
1	2	3	4	5	6

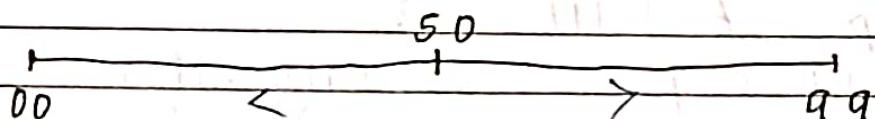
Assumed decimal pt

$$\text{Max} = +999.999$$

$$\text{Min} = -999.999$$

$$123.456 = 0.123456 \times 10^3$$

+	mantissa						$M = 0.123456$
	1	2	3	4	0	3	$\exp = 03$
	exp						$f1.pt.\exp = 50 + 03 = 53$



$$\text{Exp} = 03$$

$$f1.pt.\exp = 50 + 03 = 53$$

$$f1.pt.\exp = 50 + (-4) = 46$$

Range 0.1×10^{-50} (Min) 0.1×10^{44} (Max^m)

$$10^{-51} \text{ to } 10^{48}$$

Q. Represent 53.625 to the base 10 in binary float representation in a 16-bit register in which 9 bits form mantissa, 6 bits for exponent & 1 sign bit.

A- $(53.625)_{10} = (?)_2$

2 | 53 Mantissa tail $(110101)_2$

2 | 26 1

2 | 13 0 $0.625 \times 2 = 1.250 \rightarrow 1$

2 | 6 1 $0.25 \times 2 = 0.50 \rightarrow 0$

2 | 3 0 $0.50 \times 2 = 1.00 \rightarrow 1$

1 1

0.110101101 (110101.101)

0.110101101×2^6

2 | 50

M = 0.110101101

2 | 12 1

exp = 6 = 110

2 | 6 0

F1 pt. exp = $100000 + 110 = 100110$

2 | 3 0

100000 (32)

1 1

000000

1000111

(0)

(63)

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

sign

mantissa

exp

Parity : Counting no. of ones in a given word.

Parity
(Count)

Even	Odd
(even no. of ones)	(odd no. of ones)

0001 → 1 - 0

1110 → 2 - E

1110 → 3 - O

1111 → 4 - E

0000 → 1 - 0

1010 → 2 - E

... 9 8 7 6 5 4 3 2 1
 $D_1, P_8, D_7, D_6, D_5, P_4, D_3, P_2, P_1$

$D_7, D_6, D_5, P_4, D_3, P_2, P_1$

↓

Hamming code

Q:- Construct a hamming code with the data bit 1101.

A- $D_7 \quad D_6 \quad D_5 \quad D_4 \quad D_3 \quad P_2 \quad P_1$
 1 1 0 1

~~Even-odd~~ ~~minimum Hamming distance~~

$$P_1 = 1, 3, 5, 7$$

$$P_2 = 2, 3, 6, 7$$

$$P_4 = 4, 5, 6, 7$$

- * If The posⁿ value is even Then it is 0, if The posⁿ value is odd Then it is 1.
- $P_1 = 0$, $P_2 = 1$, $P_3 = 1$, $P_4 = 0$

Q- In a transmission The receiver got a pattern 1110110 Is There any error? If yes, Then in which data bit what is the correct data.

A-	D_7	D_6	D_5	P_4	D_3	P_2	P_1
	1	1	1	0	1	1	0
	$x = 1, 3, 5, 7$	$y = 2, 3, 6, 7$	$z = 4, 5, 6, 7$		$x = 1$	$y = 0$	$z = 1$
	P_4	P_2	P_1				
	1	0	1				
	1	0	1				
	$D_5 \rightarrow 0$						

Data $\rightarrow 1101$

- * IF $x = y = z = 0$, Then There is no error.

$x = 1, 3, 5, 7$	$y = 2, 3, 6, 7$	$z = 4, 5, 6, 7$
1	0	1
1	1	0
1	0	1
1	1	1

Gate

- 3-types :- 1) AND (Logical Multiplication)
 2) OR (Logical Add")
 3) NOT (Complement)

1) AND Gate

$$A \quad B \quad Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$A \quad B \quad Y = A \cdot B$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

2) OR Gate

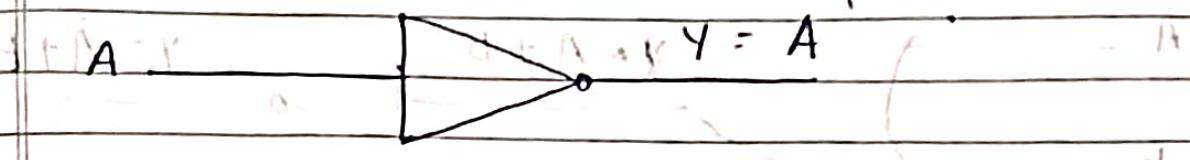
$$A \quad B \quad Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$$A \quad B \quad Y = A + B$$

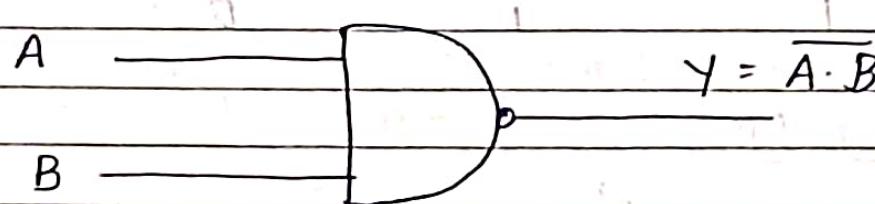
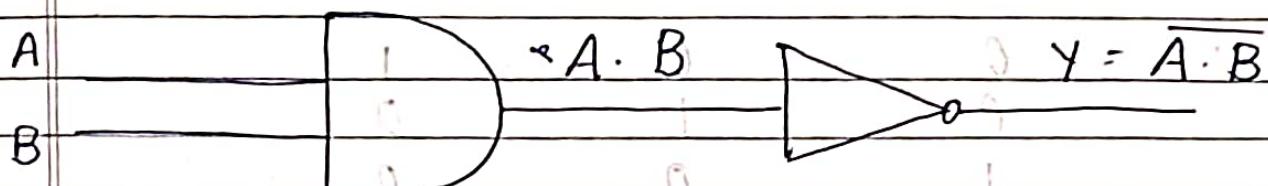
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

3) NOT Gate (Inverter)



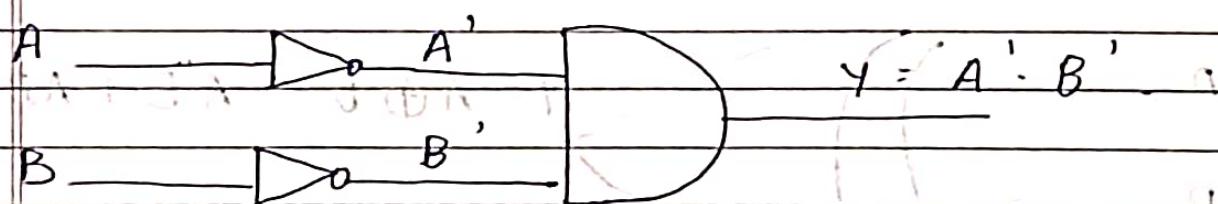
<u>A</u>	<u>$Y = A'$</u>
0	1
1	0

* NAND FA (AND + NOT)



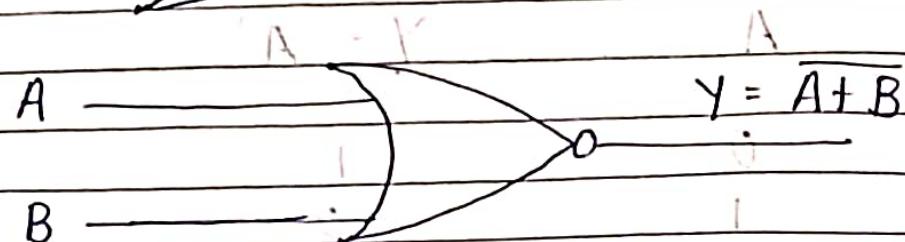
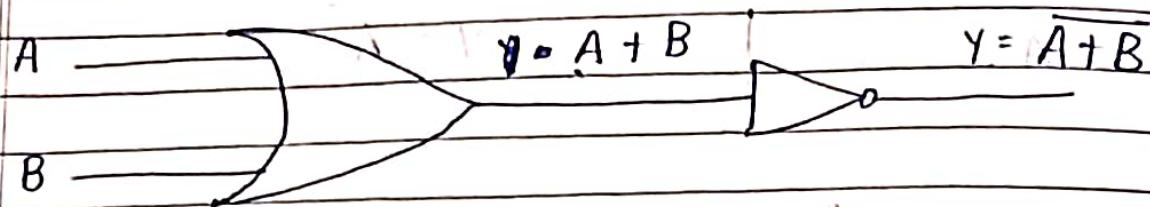
$$\underline{A} \quad \underline{B} \quad Y = \overline{A \cdot B}$$

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



BUBBLE-AND

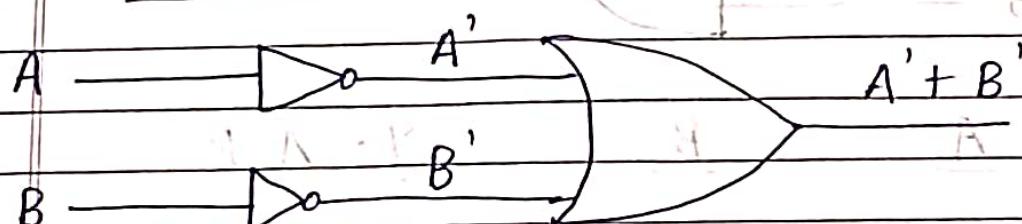
* NOR (NOT + OR) (OR + NOT)



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

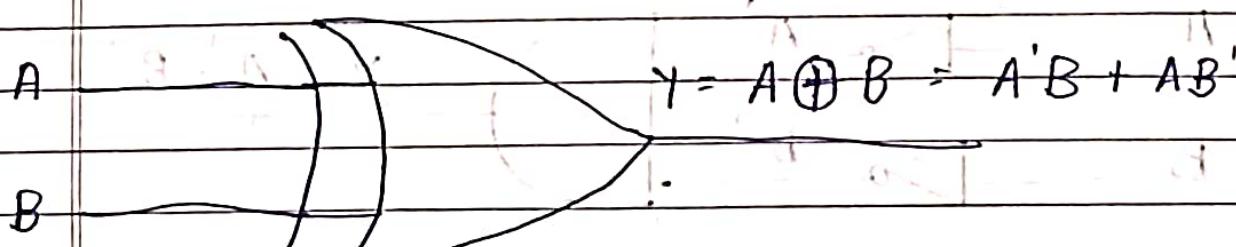
0	0	1	1
0	1	1	0
1	0	0	0
1	1	0	0

NOT + OR

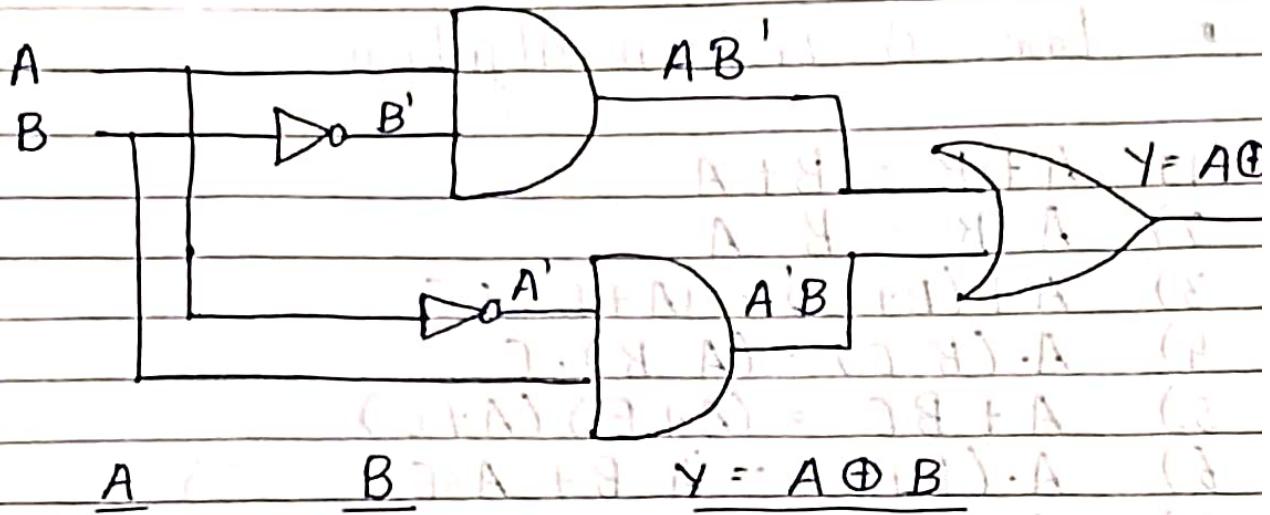


Bubble - OR

* Ex- OR / XOR



cina - 3148161



$$\underline{A} \quad \underline{B} \quad \underline{Y = A \oplus B} \quad (1)$$

$$0 \quad 0 \quad A = 0 \quad (2)$$

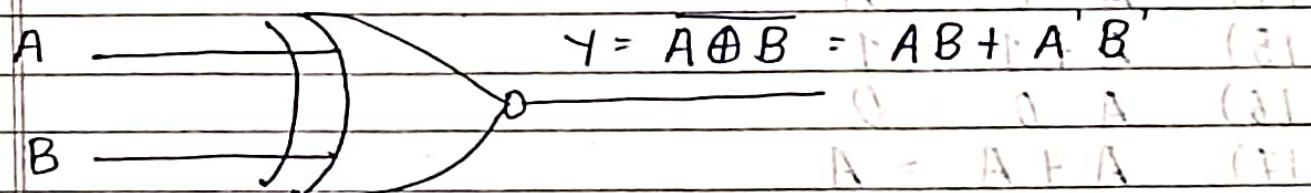
$$0 \quad 1 \quad A = 1 \quad (3)$$

$$1 \quad 0 \quad A = 1 \quad (4)$$

$$1 \quad 1 \quad A = 0 \quad (5)$$

$$(2) \cdot (3) + (4) \cdot (5) = (2+4) \cdot (3+5) \quad (6)$$

$$* \quad \underline{X - NOR} \quad \underline{A = \overline{A+A}} \quad (7)$$



$$Y = \overline{A \oplus B} = \overline{AB + A'B} \quad (7)$$

$$0 \quad 0 \quad A = 0 \quad (8)$$

$$0 \quad 1 \quad A = 1 \quad (9)$$

$$1 \quad 0 \quad A = 1 \quad (10)$$

$$1 \quad 1 \quad A = 0 \quad (11)$$

$$A = \overline{\overline{A}} = A \quad (12)$$

and similarly

$$\overline{A} \cdot \overline{A} = \overline{(A+A)} \quad (13)$$

$$\overline{A} \cdot \overline{A} = \overline{(A \cdot A)} \quad (14)$$

Laws of Boolean Algebra

- 1) $A + B = B + A$
- 2) $A \cdot B = B \cdot A$
- 3) $A + (B + C) = (A + B) + C$
- 4) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
- 5) $A + BC = (A + B)(A + C)$
- 6) $A \cdot (B + C) = A \cdot B + A \cdot C$
- 7) $A + AB = A$
- 8) $A \cdot (A + B) = A$
- 9) $A + A'B = A + B$
- 10) $A \cdot (A' + B) = AB$
- 11) $AB + A'C + BC = AB + A'C$
- 12) $(A + B) \cdot (A + C) \cdot (B + C) = (A + B) \cdot (A' + C)$
- 13) $A + 0 = A$
- 14) $A \cdot 1 = A$
- 15) $A + 1 = 1$
- 16) $A \cdot 0 = 0$
- 17) $A + A = A$
- 18) $A \cdot A = A$
- 19) $A + A' = 1$
- 20) $A \cdot A' = 0$
- 21) $A'' = A$

De-Morgan's Law

$$1) (A + B)' = A' \cdot B'$$

$$2) (A \cdot B)' = A' + B'$$

2 methods [INDUCTION]

[TRUTH TABLE]

$$A + BC = (A + B)(A + C)$$

$$\underline{LHS} \quad A + BC$$

$$= A \cdot 1 + BC \quad (\because A \cdot 1 = A)$$

$$= (A(1+B) + BC) \quad (\because 1+B = 1)$$

$$= A \cdot 1 + AB + BC \quad (1+AB = AB)$$

$$= A(1+E) + AB + BC \quad (1+E = E)$$

$$= A \cdot 1 + AC + AB + BC \quad (AC + AB = AB)$$

$$= A \cdot A + AC + AB + BC \quad (A \cdot A = A)$$

$$= A(A+C) + B(A+C) \quad (AC + BC = C(A+B))$$

$$= (A+B)(A+C) \quad (\cancel{A(A+C) + B(A+C)})$$

$$\underline{LHS} = RHS$$

$$A + A'B = A + B$$

$$\underline{LHS} \quad A + A'B$$

$$= (A + A')(A + B)$$

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

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

$$\underline{LHS} = RHS$$

E- Simplify the boolean func'

$$abc + abd + a'b'c' + cd + bd'$$

using Induction method

$$\begin{aligned}
 & abc + abd + a'bc' + cd + bd' \\
 & = abc + abd + bd' + a'bc' + cd \\
 & = abc + b(ad + d') + a'bc' + cd \\
 & = abc + b([a+d'][d+d']) + a'bc' + cd \\
 & = abc + b[(a+d') \cdot 1] + a'bc' + cd \\
 & = abc + ab + bd' + a'bc' + cd \quad \text{DII} \\
 & = ab(c+1) + bd' + a'bc' + cd \\
 & = ab + a'bc' + bd' + cd \\
 & = ab + b(c+a'c') + bd' + cd \\
 & = ab + b(1 + a'c') + bd' + cd \\
 & = ab + b(a+c') + bd' + cd \\
 & = ab + b(a+c') + bd' + cd \\
 & = ab + b(a+c') + bd' + cd \\
 & = ab + b(c'+d') + cd \\
 & = ab + b\overline{cd} + cd \quad \text{DII} \\
 & = ab + (b+cd)(cd + \overline{cd}) \\
 & = ab + (b+cd) \quad \text{d'A = R A + A} \\
 & = ab + b + cd \\
 & = b(a+1) + cd \quad \text{S'A + A} \quad \text{DII} \\
 & = b + cd \quad \cancel{(S+A)(A+A)} \\
 & \quad \cancel{(S+A)P} \\
 & \quad \cancel{S+A} \\
 & \quad \cancel{R} \\
 & \quad \cancel{R} = \cancel{R}
 \end{aligned}$$

∴ $b + cd$ is the simplified expression.

$(atb)(atc)$ a b c bc $atbc$ atb atc

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

$a^2 b^2 c^2$

$$(atb)^2 = ?$$

$$(atb + ct \dots + nt)^2$$

$$= a^2 \cdot b^2 \cdot c^2 \dots n^2$$

$$\text{Let } atb = x$$

$$(x + r)^2$$

$$= x^2 + c^2$$

$$= (A + B)^2 \cdot C^2$$

$$= A^2 \cdot B^2 \cdot C^2$$

$$(a \cdot b \cdot c)^2 = ?$$

$$\text{Let } b \cdot c = x$$

$$(a \cdot x)^2$$

$$= a^2 + x^2$$

$$= (b \cdot c) + a^2$$

$$= a^2 + b^2 + c^2$$

$$(a \cdot b \cdot c \dots n)^2$$

$$= a^2 + b^2 + c^2 + \dots n^2$$

Find the complement of the function
 $(a'b + ab')$

$$A = (a'b)' \cdot (ab')'$$

$$= (\bar{a} + b') \cdot (a' + \bar{b})$$

$$= (a + b') \cdot (a' + b)$$

$$= a \cdot a' + a \cdot b + b \cdot a' + b \cdot b$$

$$= 0 + ab + ba' + 0$$

$$= ab + a'b'$$

* Min Term & Max Term

(AND)

(OR)

a	b	c	Minterm	Maxterm
Term	Desg	Term	Desg	
0 0 0	$a'b'c'$	m_0	$a+b+c$	M_0
0 0 1	$a'b'c$	m_1	$a+b+c'$	M_1
0 1 0	$a'b'c$	m_2	$a+b+c$	M_2
0 1 1	$a'b'c$	m_3	$a+b+c'$	M_3
1 0 0	$a'b'c'$	m_4	$a'+b+c$	M_4
1 0 1	$a'b'c$	m_5	$a'+b+c'$	M_5
1 1 0	$a'b'c$	m_6	$a'+b+c$	M_6
1 1 1	$a'b'c$	m_7	$a'+b+c'$	M_7

0 → complement } Minterm
 1 → actual } Maxterm

0 → actual

1 → complement } Maxterm

Sum of minterms $\rightarrow a'b'c + a'b'c'$

Product of maxterms $\rightarrow (a+b+c) \cdot (a+b+c')$

Q- Represent $f = ab'c$ in sum of minterm form.

$$A- \quad F = ab'c$$

$$a = a \cdot 1$$

$$= a(b+b')$$

$$= ab + ab' \quad \text{--- } ①$$

$$ab = ab \cdot 1$$

$$= ab(c+c')$$

$$\Rightarrow ab = abc + abc' \quad \text{--- } ②$$

$$ab' = ab' \cdot 1$$

$$= ab'(c+c')$$

$$= ab'c + ab'c' \quad \text{--- } ③$$

$$b'c = b'c \cdot 1$$

$$= b'c(a+a')$$

$$= ab'c + a'b'c \quad \text{--- } ④$$

By combining eqⁿ 2, 3, 4

$$F = abc + abc' + ab'c + ab'c' + ab'c + a'b'c$$

$$\Rightarrow F = abc + abc' + ab'c + ab'c' + a'b'c$$

$$= m_7 + m_6 + m_5 + m_4 + m_1$$

$$\Rightarrow F(a, b, c) = \sum (1, 4, 5, 6, 7)$$

Q- Represent $F(a, b, c) = 1$ in sum of minterm form.

A- $F(a, b, c) = 1$

$$\begin{aligned}
 &= a + a' \\
 &= a(b + b') + a'(b + b') \\
 &= ab + ab' + a'b + a'b' \\
 &= a'b'(c + c') + ab(c + c') + ab'(c + c') + a'b(c + c') \\
 &= a'b'c + a'b'c' + abc + abc' + ab'c + ab'c' \\
 &= m_7 + m_6 + m_5 + m_4 + m_3 + m_2 + m_1 + m_0
 \end{aligned}$$

$$\therefore f(a, b, c) = \sum(0, 1, 2, 3, 4, 5, 6, 7)$$

Q- Represent $f = xy + x'z$ in product of maxterm form.

$$\begin{aligned}
 f &= xy + x'z \\
 &= (x'y + x'z)(xy + z) \\
 &= (x + x')(x'y + y)(x + z)(y + z) \\
 &= (x'y + y)(x + z)(y + z) \quad \text{--- } ①
 \end{aligned}$$

$$\begin{aligned}
 x'y + y &= x'y + y + 0 \\
 &= x'y + y + z \cdot z' \\
 &= (x'y + y + z)(x'y + y + z') \quad \text{--- } ②
 \end{aligned}$$

$$\begin{aligned}
 x + z &= x + z + 0 \\
 &= x + z + y \cdot y' \\
 &= (x + y + z)(x + y' + z) \quad \text{--- } ③
 \end{aligned}$$

$$\begin{aligned}
 y+z &= \bar{x} + y + z \\
 &= \bar{x}' \cdot \bar{x} + y + z \\
 &= (\bar{x}' + y + z) (x + y + z) \quad \text{--- (4)}
 \end{aligned}$$

$$\begin{aligned}
 f &= (x' + y + z) (x' + y + z') (x + y + z) (x + y' + z) \\
 &\quad (\bar{x}' + y + z) (\bar{x} + y + z) \\
 \Rightarrow f &= (\bar{x}' + y + z) (\bar{x}' + y + z') (x + y + z) \\
 &\quad + (x + y' + z) \\
 \Rightarrow f &= M_4 \cdot M_5 \cdot M_0 \cdot M_2
 \end{aligned}$$

$$\therefore f(x, y, z) = \pi(0, 2, 4, 5)$$

Q- Represent $f(x, y, z) = 0$ in product of maxterm form

$$\begin{aligned}
 \underline{\text{A-}} \quad f(x, y, z) &= 0 \\
 &= \bar{x} \cdot \bar{x}' \\
 &= \bar{x}(1 + 0) \cdot \bar{x}'(1 + 0) \\
 &= \bar{x}(y \cdot y') \cdot \bar{x}'(y' + y \cdot y') \\
 &= (\bar{x} + y)(\bar{x} + y')(x' + y)(x' + y') \\
 &= 1
 \end{aligned} \quad \text{--- (1)}$$

$$\begin{aligned}
 \bar{x} + y &= \bar{x} + y + 0 \\
 &= \bar{x} + y + z \cdot z' \\
 &= (\bar{x} + y + z)(\bar{x} + y + z') \quad \text{--- (2)}
 \end{aligned}$$

$$\begin{aligned}
 \bar{x} + y' &= \bar{x} + y' + 0 \\
 &= \bar{x} + y' + z \cdot z' \\
 &= (\bar{x} + y' + z)(\bar{x} + y' + z') \quad \text{--- (3)}
 \end{aligned}$$

$$\begin{aligned}
 \bar{x}' + y &= \bar{x}' + y + 0 \\
 &= \bar{x}' + y + z \cdot z' \\
 &= (\bar{x}' + y + z)(\bar{x}' + y + z') \quad \text{--- (4)}
 \end{aligned}$$

$$\begin{aligned}
 (x'y') &= x'y' + 0 \\
 &= x'y' + (z \cdot z') \\
 &= (x'y' + z) (x'y' + z') \quad (5)
 \end{aligned}$$

Combining eqⁿ 2, 3, 4, 5

$$\begin{aligned}
 f &= (x+y+z)(x+y+z')(x+y+z)(x+y+z') \\
 &\quad (x'+y+z)(x'+y+z')(x'+y+z')(x'+y+z')
 \end{aligned}$$

$$\Rightarrow f = M_0 \cdot M_1 \cdot M_2 \cdot M_3 \cdot M_4 \cdot M_5 \cdot M_6 \cdot M_7$$

$$\therefore f(x, y, z) = \pi(0, 1, 2, 3, 4, 5, 6, 7)$$

Karnaugh Map (K-Map)

~~SOP~~ 2 variable

x\y	0	1	0	1
0	0 → m ₀	1	x'y'	x'y
1	y → m ₁	0	xy'	xy
	1 → m ₂			
	1 → m ₃			

Q- $f(x, y) = \sum(1, 2)$

x\y	0	1	0	1
0	0	1	0	1
1	1	0	2	3

$$f = xy' + x'y$$

Q- $f(x, y) = \sum (1, 2, 3)$

$x \backslash y$	0	1	1	1
0	0	1	1	1
1	1	2	1	3

$$f = x + y$$

3 variable

$x \backslash y \backslash z$	000	001	011	101	110
0	0	0	1	3	2
0	0	1	1	5	7
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Q- $f(x, y, z) = \sum (2, 3, 4, 5)$

$x \backslash y \backslash z$	00	01	11	10
0	0	1	1	2
1	4	5	7	6

$$f = xy' + x'y$$

$$\begin{aligned}
 &= x(y'z' + y'z) \\
 &= x[y'(z' + z)] \\
 &= xy
 \end{aligned}$$

Q- $f(x, y, z) = \sum (0, 2, 4, 6)$

$x \backslash yz$	00	01	101	10
0	10	1	3	12
1	14	5	7	16

$$\begin{aligned}
 f &= y'z' + yz \\
 &= z'(y' + y) \\
 &= z
 \end{aligned}$$

Q- $f(x, y, z) = \sum (1, 2, 3, 4, 5, 6, 7)$

$x \backslash yz$	00	01	11	10
0	0	11	13	12
1	14	15	17	16

$$f = y'z + x$$

$$f = x + y + z$$

$$\begin{aligned}
 &1(00+01+11+10) \\
 &= x(y'z' + y'z + yz + y'z') \\
 &= x[y(z+z) + y'(z+z')] \\
 &= x(y + y') \\
 &= x
 \end{aligned}$$

Q- $f(x, y, z) = \sum (0, 3, 5, 6)$

$x \backslash yz$	00	01	11	10
0	1		1	
1		1		1

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

4 variable Karnaugh Map (hard)

- 1) One square represent one minterm which will produce 4 literal terms
- 2) Two consecutive square will produce 3 literal terms
- 3) Four consecutive square will produce 2 literal terms.
- 4) 8 consecutive square will produce 1 literal terms.
- 5) 16 consecutive square will give functional value equal to 1.

$ab \backslash cd$	00	01	11	10
00				
01				
11				
10				

Q-1 $f(a, b, c, d) = \sum (0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$

$ab \backslash cd$	00	01	10	11	00	01	10	11
00	1	0	1	0	1	2	1	2
01	1	1	1	1	1	1	1	1
11	1	0	1	1	1	1	1	1
10	1	1	0	1	1	1	1	0

$$f = c' + a'd' + bd'$$

Q-2 $f(a,b,c,d) = \sum (0, 2, 3, 5, 8, 10, 11)$

		cd	00	01	11	10	
ab	00	00	D		I	I	
		01		(I)			
ab	11	11			I	(I)	I
		10	D		I	(I)	I

$$f = b'd' + b'c + a'b'c'd'$$

DG

4 variable

Q- $f(a,b,c,d) = \pi (0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$

		cd	00	01	11	10	
ab	00	00	0	0		0	
		01	0	0		0	
ab	11	11	0	0		0	
		10	0	0			

$$\bar{f} = c' + a'd' + bd'$$

$$(M.S.) \Rightarrow f_D = c(a+d)(b'+d) \quad (\text{Change in } f)$$

Don't care cond'

Q- $f(a,b,c,d) = \sum (1, 3, 7, 11, 15)$

$$d(a,b,c,d) = \sum (0, 2, 5)$$

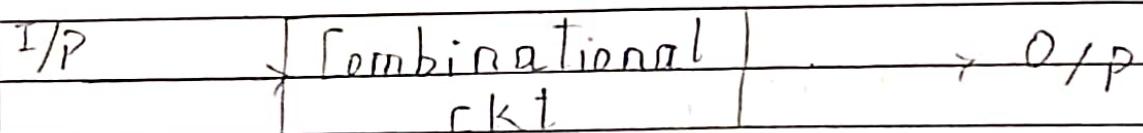
Find SOP & POS.

ab \ cd	00	01	11	10	
00	X	1	1	X	
01	0	X	1	0	
11	0	0	1	0	
10	0	0	1	0	

SOP $F = cd + a'd$

SOP $\begin{cases} F = d'(a' + c) \\ F' = d' + a'c \end{cases}$
 $\Rightarrow F = d(a' + c)$

20/07/2nd Combinational Circuit



- 1) Adder
- 2) Subtractor
- 3) Multiplexer
- 4) De multiplexer
- 5) Decoder
- 6) Encoder

1) Adder

It can perform only addition.

- i) Half - Adder
- ii) Full - Adder

i) Half Adder

Only 2-bit of addition can be done
in half-adder

I/P		O/P	
<u>x</u>	<u>y</u>	<u>S</u>	<u>C</u>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
$x \setminus y$		D	I
SOP	00	(1)	$S' + y + xy'$
	1	(1)	=

$$S = \overline{x}y + \overline{x}y'$$

\overline{x}	y	0	1
0	0	(0)	
1	0	(0)	

$$S' = \overline{x}y' + xy$$

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

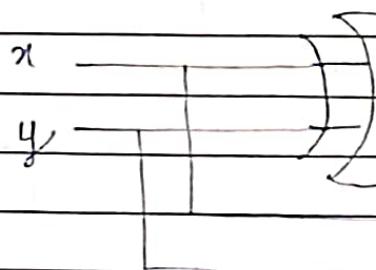
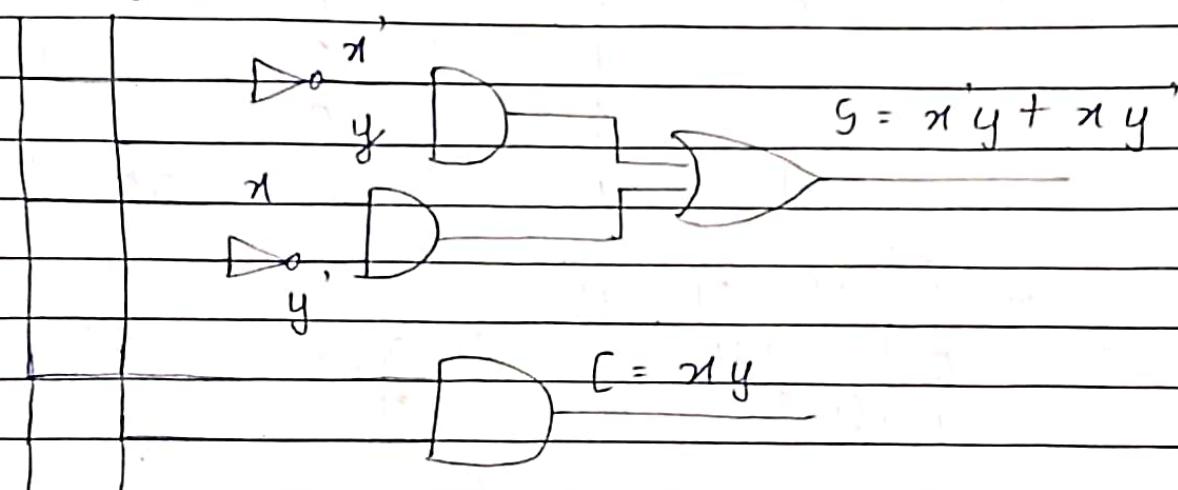
x	y	0	1
-----	-----	---	---

0			
1		(1)	

$$C = xy$$

$$\begin{aligned} S &= x'y + xy' \rightarrow x \oplus y \\ C &= xy \end{aligned}$$

x	y
-----	-----



Half - Adder

ii) Full Adder

Only 3-bit of addition can be done in full adder.

I.P.

O.P.

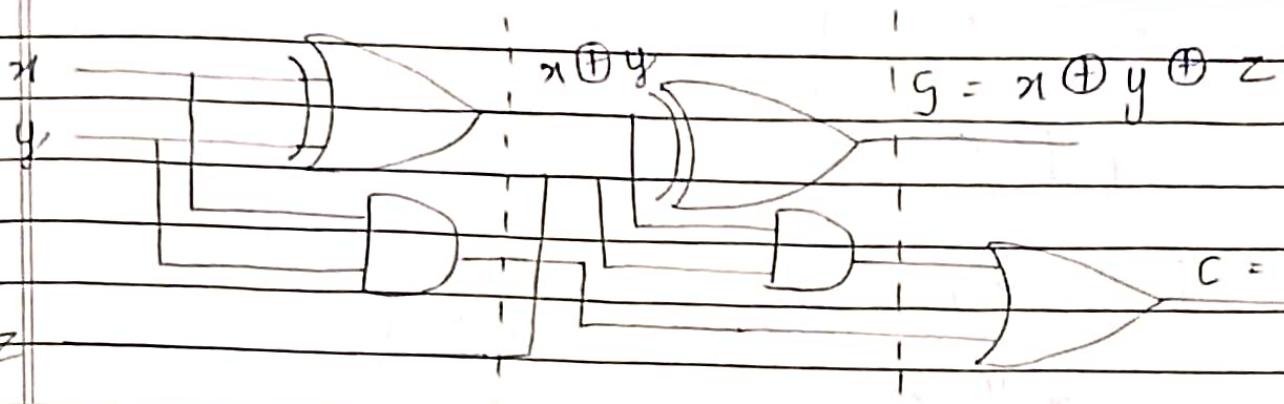
<u>21</u>	<u>41</u>	<u>Z</u>	<u>S</u>	<u>C</u>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

<u>x</u> \ <u>y z</u>	00	01	11	10
0	0	1	3	2
1	4	5	7	6

$$\begin{aligned}
 S &= x'y'z + x'yz' + xy'z' + xyz \\
 &= x'(y'z + yz') + x(y'z' + yz) \\
 &= x'(y \oplus z) + x(y \oplus z) \\
 &= x \oplus y \oplus z
 \end{aligned}$$

<u>x</u> \ <u>y z</u>	00	01	11	10
0			1	
1		1	11	1

$$C = xy + xz + yz$$



Q- Design a full adder using half adders only.

Subtractors

⇒ It is a combinational circuit which performs subtraction.

2 types :- i) Half
ii) Full

i) Half Subtractors

It is a combinational / arithmetic circuit which can perform only 1 bit of subtraction.

I/P	O/P
<u>x</u>	<u>y</u>
0	0
0	1
1	0
1	1

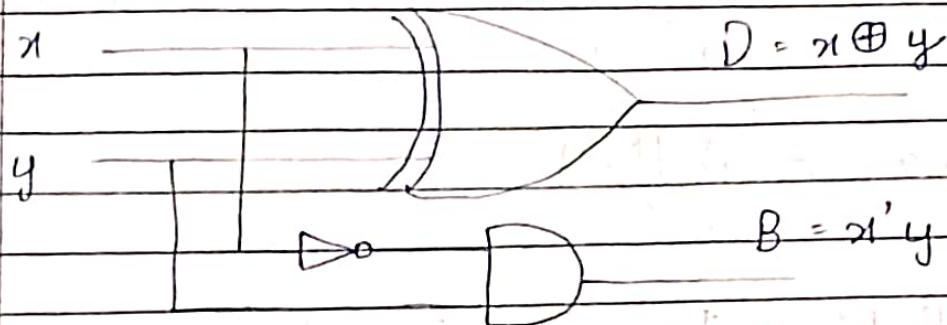
<u>x</u>	<u>y</u>	<u>D</u>	<u>B</u>
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

x	y	0	1	
0			0	
1		0		

$$D = \bar{x}'y + xy' = x \oplus y$$

x	y	0	1	
0			0	
1		0		

$$B = \bar{x}'y$$



Half Subtractor

iii) Full Subtractor

3 bit of subtraction can be taken place in full subtractor.

<u>x</u>	<u>y</u>	<u>z</u>	<u>D</u>	<u>B</u>
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0

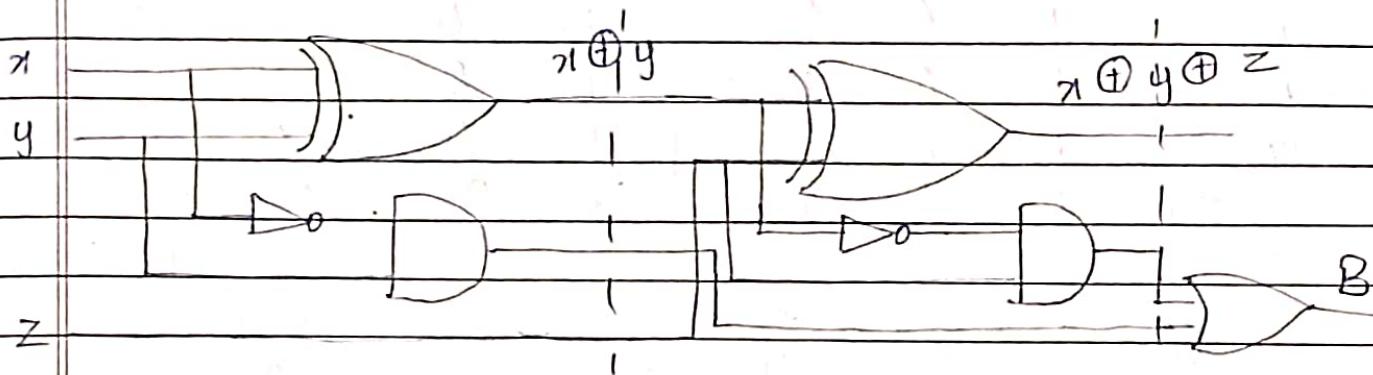
1	0	1	-	0	0	1	1	0
1	1	0	-	0	0	1	0	1
1	1	1	-	1	1	1	1	1

$x \setminus y \setminus z$	00	01	11	10	
0	0	1	1	1	
1	1	0	0		

$$\begin{aligned} D &= x'y'z + x'yz' + xy'z' + xyz \\ &= x' \oplus y \oplus z \end{aligned}$$

$x \setminus y \setminus z$	00	01	11	10	
0	0	1	1	1	
1	1	0	0		

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



Full Subtractor

Design a full subtractor with half subtractors only.

Q-1

Design a 4 bit binary to gray code converter.

O/P

A-

	B_3	B_2	B_1	B_0		G_{13}	G_{12}	G_1	G_0
0 \rightarrow	0	0	0	0		0	0	0	0
1 \rightarrow	0	0	0	1		0	0	0	1
2 \rightarrow	0	0	1	0		0	0	1	1
3 \rightarrow	0	0	1	1		0	0	1	0
4 \rightarrow	0	1	0	0		0	1	1	0
5 \rightarrow	0	1	0	1		0	1	1	1
6 \rightarrow	0	1	1	0		0	1	0	1
7 \rightarrow	0	1	1	1		0	1	0	0
8 \rightarrow	1	0	0	0		1	1	0	0
9 \rightarrow	1	0	0	1		1	1	0	1
10 \rightarrow	1	0	1	0		1	1	1	1
11 \rightarrow	1	0	1	1		1	1	1	0
12 \rightarrow	1	1	0	0		1	0	1	0
13 \rightarrow	1	1	0	1		1	0	1	1
14 \rightarrow	1	1	1	0		1	0	0	1
15 \rightarrow	1	1	1	1		1	0	0	0

 B_0 G_{13} (LSB) B_1 G_{12} B_2 G_1 B_3 G_0 (MSB)

(4-bit binary to Gray code)

$B_3\ B_2$	$B_1\ B_0$	00	01	11	10
B_3					
00					
01		1	1	1	1
11	1	1	1	1	1
10	1	1	1	1	1

$$G_{13} = B_3$$

$B_3\ B_2$	$B_1\ B_0$	00	01	11	10
B_3					
00					
01		1	1	1	1
11					
10	1	1	1	1	1

$$G_{12} = B_3' B_2 + B_3 B_2'$$

$$= B_3 \oplus B_2$$

$B_3\ B_2$	$B_1\ B_0$	00	01	11	10
B_3					
00					
01		1	1	0	0
11	1	1	0	0	0
10	1	0	1	1	1

$$G_{11} = B_2 B_1' + B_2' B_1$$

$$= B_2 \oplus B_1$$

$B_3\ B_2$	$B_1\ B_0$	00	01	11	10
B_3					
00					
01		1	0	1	1
11	1	0	0	1	1
10	1	1	0	1	1

$$G_{10} = B_1' B_0 + B_1 B_0'$$

$$= B_1 \oplus B_0$$

Q- Design a 4 bit gray to binary code inverter.

I/P

O/P

G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
-------	-------	-------	-------	-------	-------	-------	-------

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

G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
-------	-------	-------	-------	-------	-------	-------	-------

00							
01							
11	1	1	1	1			
10	1	1	1	1			

$$B_3 = G_3$$

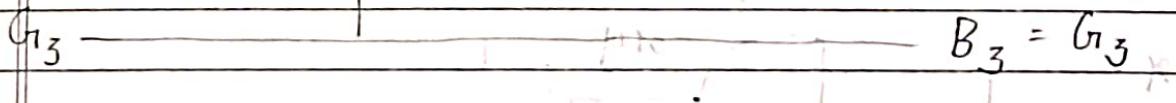
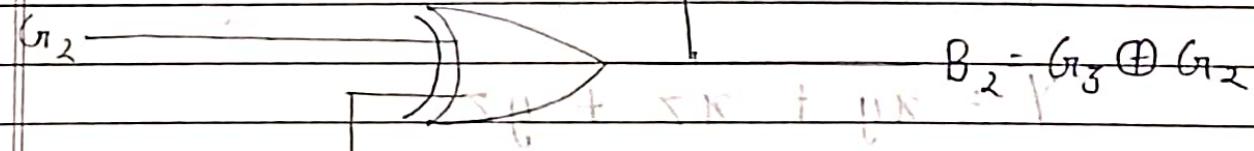
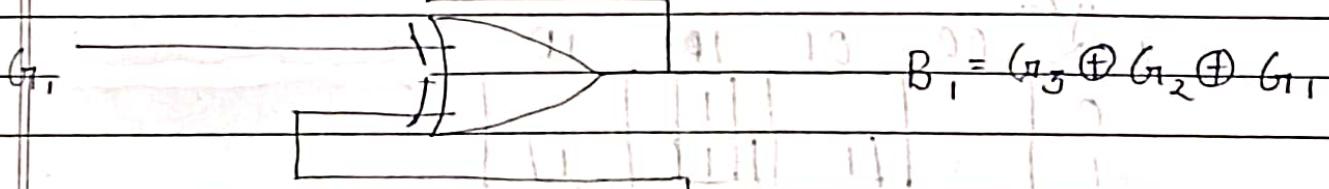
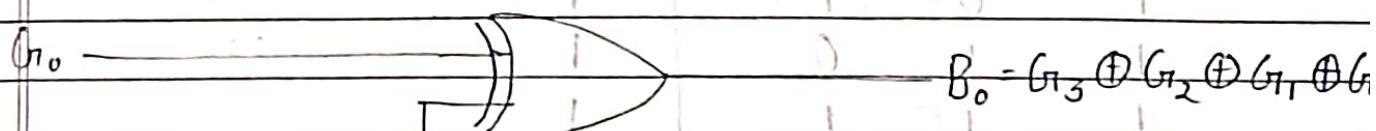
G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
00							
01	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1

$$B_2 = G_3' G_2 + G_3 G_2'$$

$$= G_3 \oplus G_2$$

		00	01	11	10	$B_1 = G_3' G_2' G_1 + G_3' G_2 G_1' + G_3 G_2' G_1 + G_3 G_2 G_1'$
$G_3 G_2$	00	1	1	1	1	$= G_3 \oplus G_2 \oplus G_1$
01	1	1	1	1		
11			1	1		
10	1	1	1	1		

		00	01	11	10	$B_0 = G_3' G_2' G_1' G_0 + G_3' G_2' G_1 G_0' + G_3' G_2 G_1' G_0 + G_3' G_2 G_1 G_0' + G_3 G_2' G_1' G_0 + G_3 G_2' G_1 G_0' + G_3 G_2' G_1' G_0' + G_3 G_2' G_1 G_0$
$G_3 G_2$	00	1	1	1	1	$= G_3 \oplus G_2 \oplus G_1 \oplus G_0$
01	1	1	1	1		
11		1	1	1		
10	1	1	1	1		



26/09/24

Page No. _____

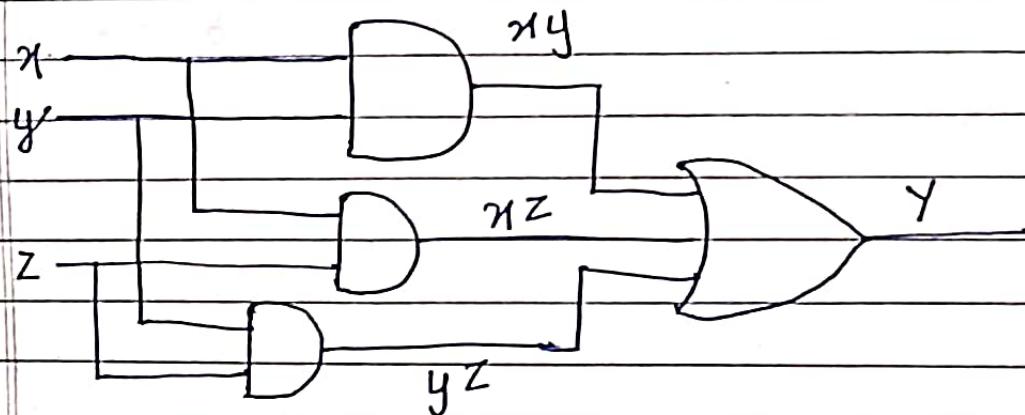
Date / /

Q-1 A majority funcⁿ is generated in a combinational circuit when the output is equal to 1 if the input variables have more no. of 1 than 0. The output is 0 otherwise. Design a 3 input majority func.

A-	x	y	z		y
	0	0	0		0
	0	0	1		0
	0	1	0		0
	0	1	1		1
	1	0	0		0
	1	0	1		1
	1	1	0		1
	1	1	1		1

x \ y \ z	00	01	101	110
0	0	0	1	0
1	1	1	1	1

$$Y = xy + xz + yz$$



Q-2 Design a combinational circuit with 3 inputs x, y, z & 3 outputs A, B, C. When the binary input is 0, 1, 2, 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, 7, the binary output is 1 less than the input.

Y/P O/P

A	x	y	z	a	b	c
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	1	0	1	0	0
1	1	0	0	1	0	1
1	1	1	1	1	1	0

x \ y \ z	00	01	11	10
0	0	1	1	0
1	1	0	0	1

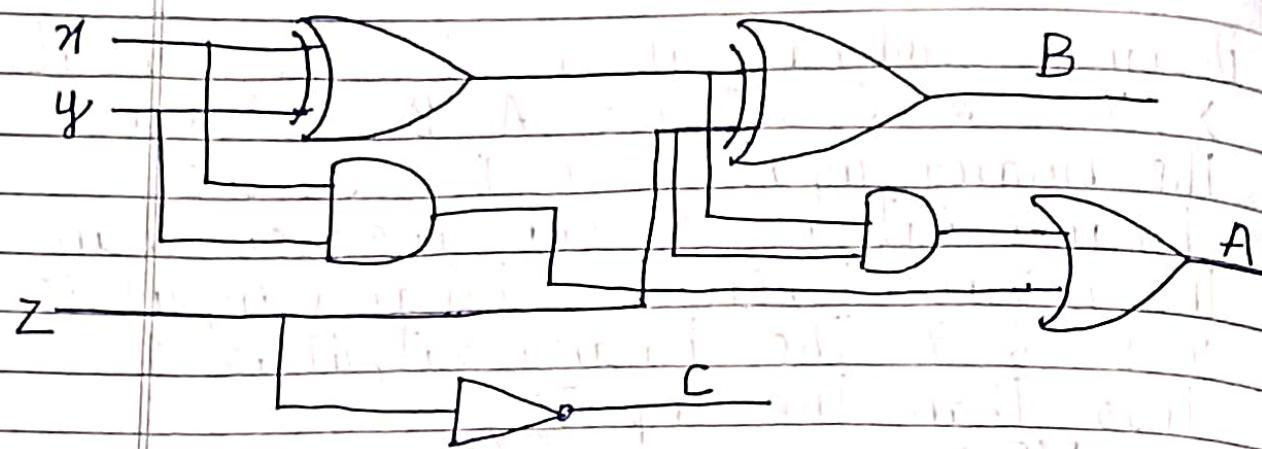
$$a = xy + yz + xz$$

x \ y \ z	00	01	11	10
0	0	1	1	0
1	1	0	0	1

$$\begin{aligned} b &= xyz' + x'y'z + xy'z' \\ &= x \oplus y \oplus z \end{aligned}$$

x \ y \ z	00	01	11	10
0	1	1		1
1	1	1		1

$$\begin{aligned} c &= y'z' + yz' \\ &\Rightarrow z \end{aligned}$$

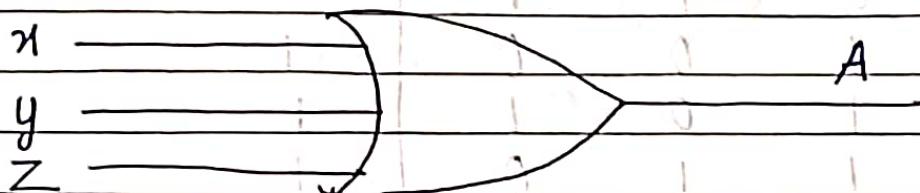


Q-3 The staircase light of a 3 storied building occupied by 3 tenants is to be switched on/off independently by them using switches located at the entrance of their flat. Design a combinational circuit of this situation.

A-	X	Y	Z	A
	0	0	0	0
	0	0	1	01
Sum of line = 0	0	1	1	11
	0	1	1	11
	1	0	0	1
	1	0	1	01
Sum of line = 0	1	1	0	11
Sum of line = 1	1	1	1	01

$x \setminus y \setminus z$	00	01	11	10	011	001	010	000
0	1	1	1	1	-	-	-	-
1	1	1	1	1	1	1	1	1

$$A = x + y + z$$



Q-4 4 students (S_1, S_2, S_3, S_4) with diff. weightage of voting take part in deciding the exam schedule. The weightage assigned to S_1, S_2, S_3 & S_4 are 30%, 40%, 20%, 10% respectively. Any combination of voting which will result into either 70% or more than 70% is acceptable by the university.

Prepare a circuit diagram to represent above cond?

\sum	S_1	S_2	S_3	S_4	y
0	0	0	0	0	0
0	0	0	0	1	0
0	0	1	0	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	1	1

(30) (40) (20) (10)

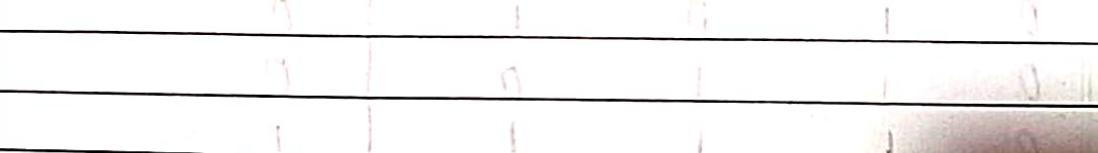
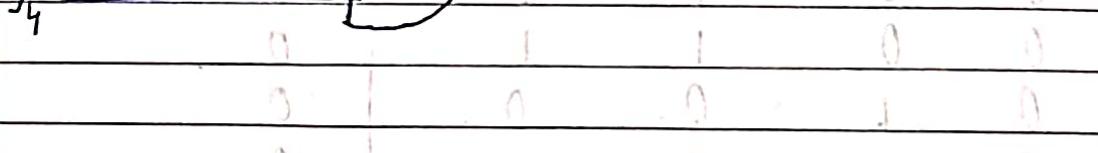
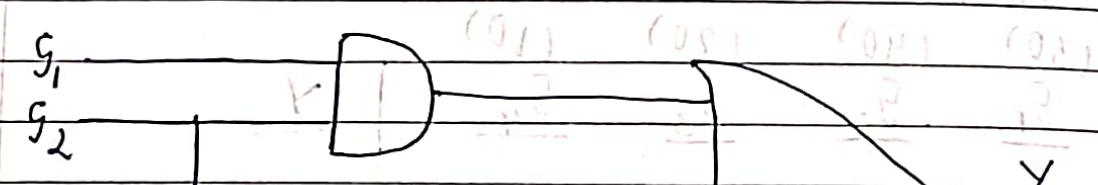
g_1 g_2 g_3 g_4

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

 \rightarrow

$g_1 g_2$	$g_3 g_4$	00	01	11	10
00					
01				1	
11		1	1	1	1
10					

$$Y = g_1 g_2 + g_2 g_3 g_4$$

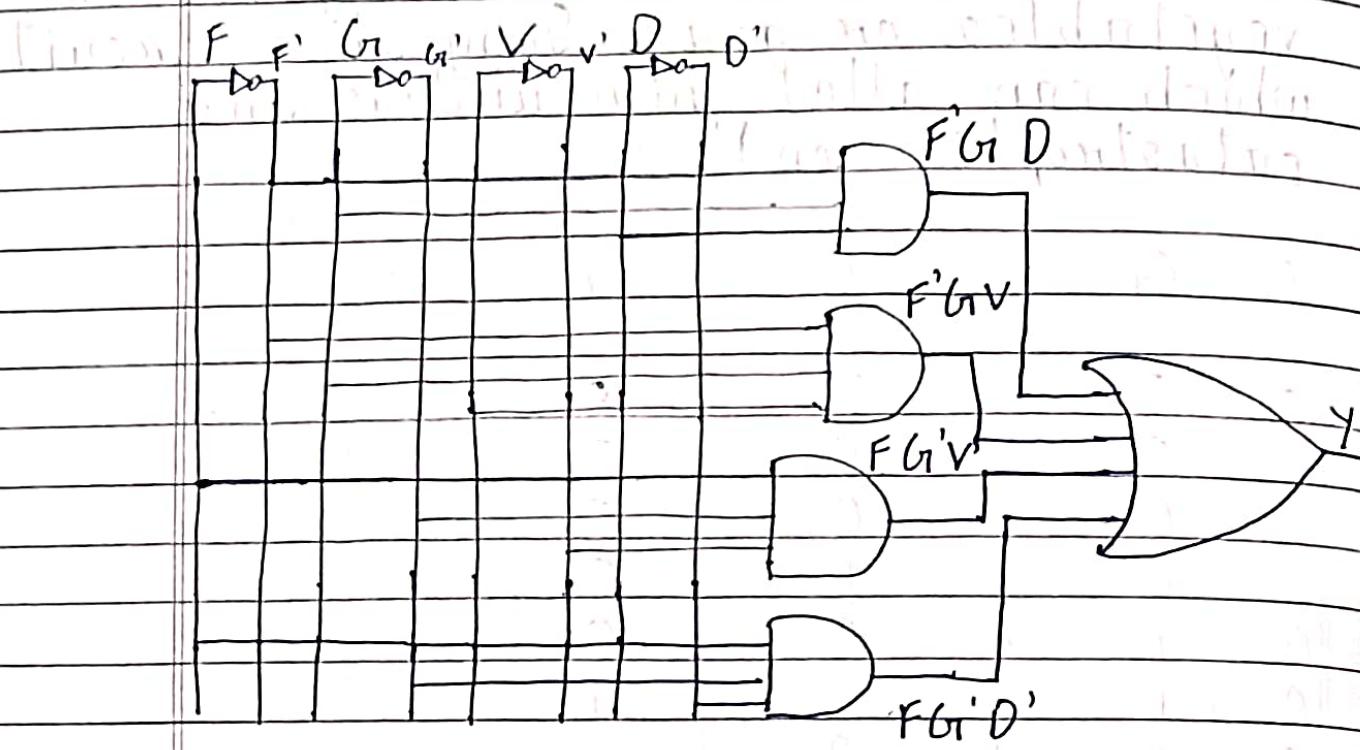


Q-1 A farmer (F) has 2 sheds (N & S) in his farm to keep farm produced & belonging. He keeps the vegetables (V) as farm produced, a goat (G) which can eat the vegetables & a dog (D) which can bite the goat in the absence of farmer. The farmer alone has to manage both the shed without loosing either vegetables or goat. Suggest a circuit which can allot him under any catastrophic cond".

A-	F	G	V	D	Y
0	0	0	0	0	0
0	0	0	0	1	0
0	0	1	0	0	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	0	0
1	1	1	1	0	1

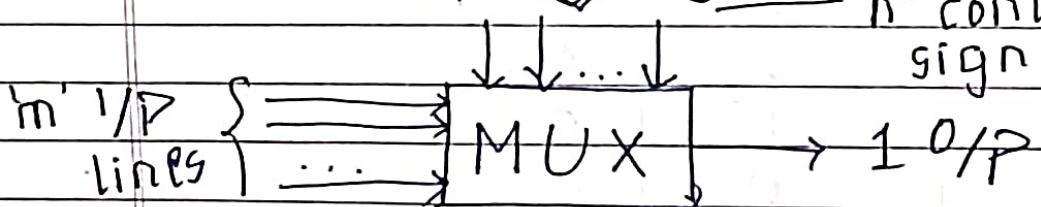
$F'G_1$	00	01	11	10	0
00					
01		1	1	1	
11					
10	1	1		1	

$$Y = F'G_1 D + F'G_1 V + FG_1 V' + FG_1 D'$$

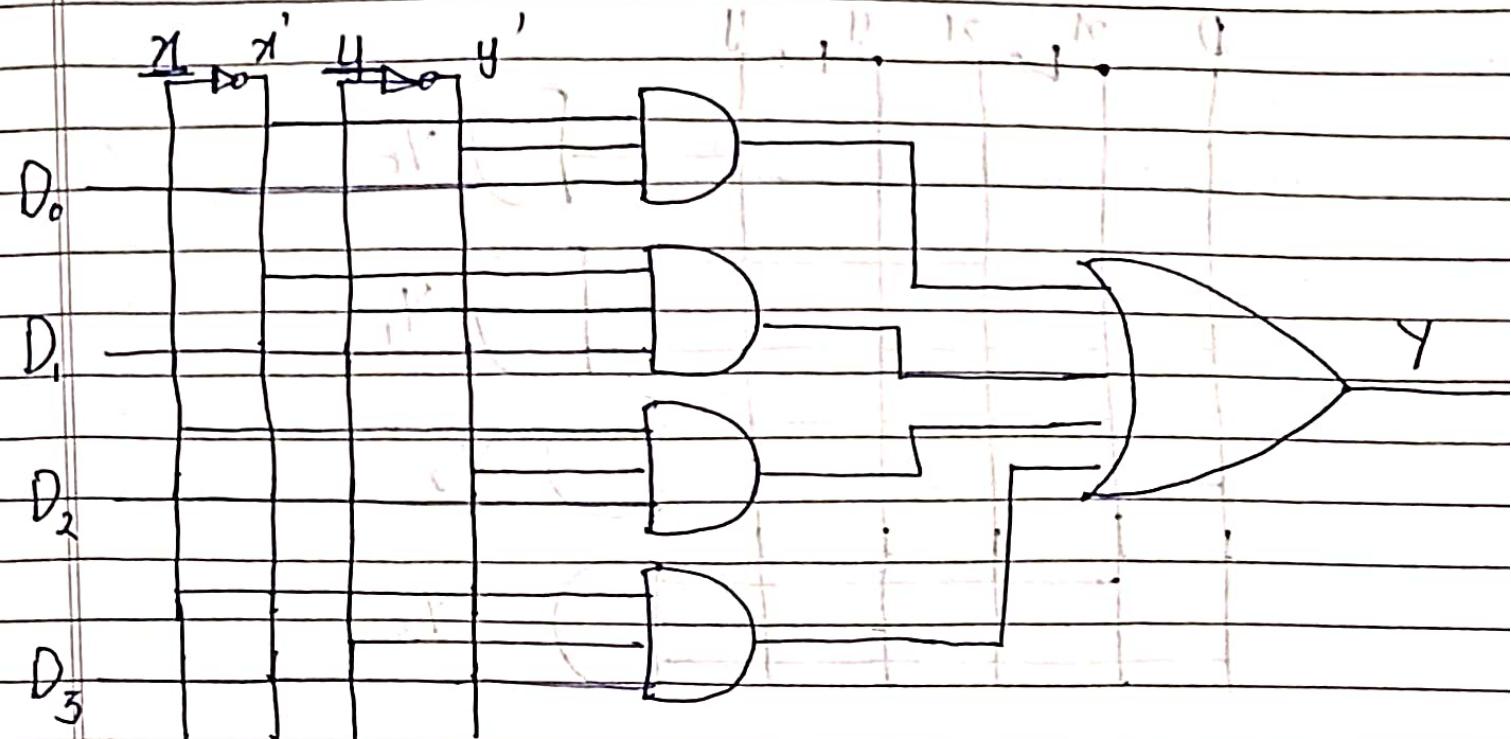


3) Multiplexer

Multiplex means many to one.
i.e. many inputs but one output
in control signals (1 to n)



4:1 MUX



$$x = 0, y = 0, Y = D_0$$

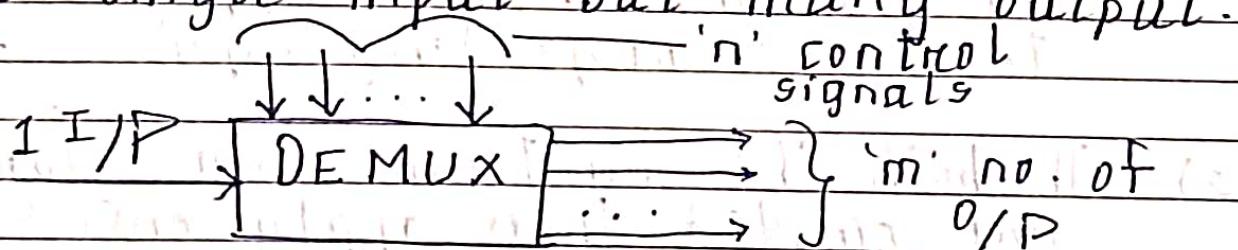
$$x = 1, y = 0, Y = D_2$$

$$x = 0, y = 1, Y = D_1$$

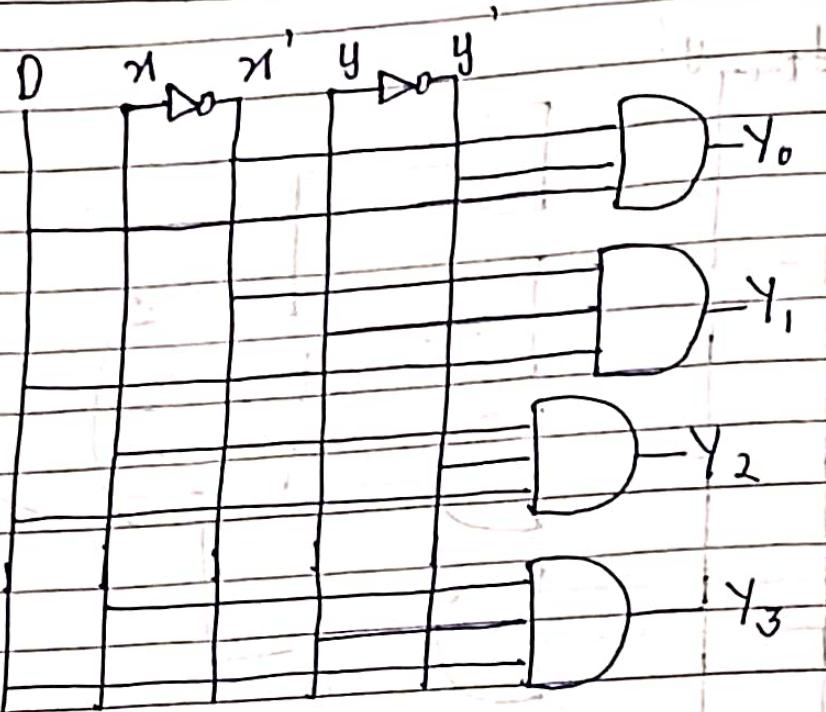
$$x = 1, y = 1, Y = D_3$$

Demultiplexer

Demultiplex means one to many i.e. single input but many output.



1) 1:4 Demux



$$x = 0, y = 0, D = Y_1$$

$$x = 0, y = 1, D = Y_2$$

$$x = 1, y = 0, D = Y_3$$

$$x = 1, y = 1, D = Y_4$$

~~30/09/24~~ Implementation of Boolean Func' using Multiplexers

1) If both The minterms in a column are not circled Then apply zero to the corresponding input.

2) If both The minterms in a column are circled Then apply one to the corresponding input.

If there are n variables,
 $\rightarrow (n-1)$ MUX

Page No. _____

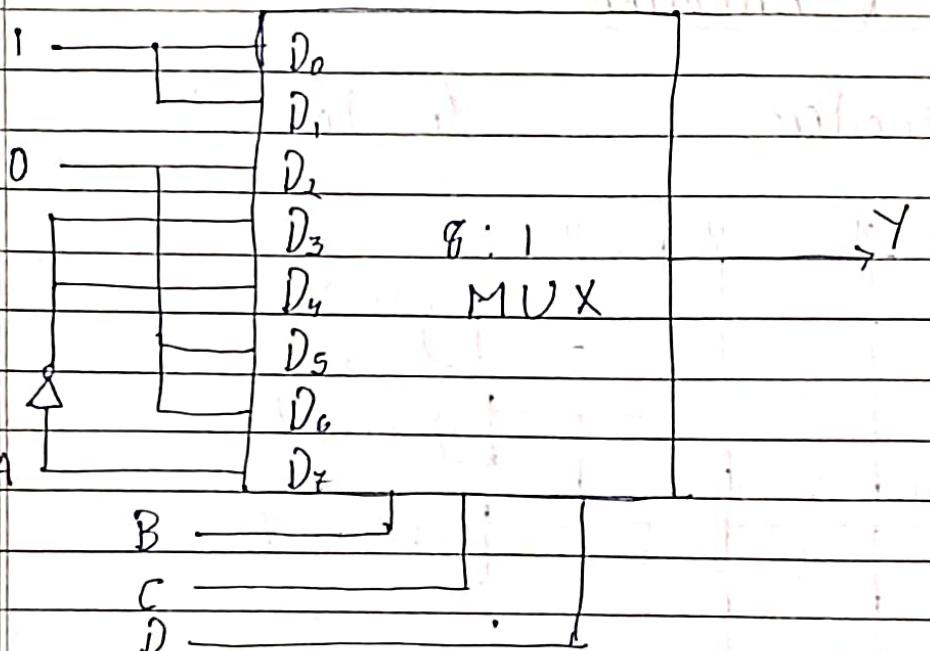
Date: / /

3) If the bottom min term is circled & The top is not circled Then apply actual value (A) to the input.

1) If The top min term is circled & The bottom is not circled Then apply complement value (A') to the input.

2- $f(A, B, C, D) = \sum(0, 1, 3, 4, 8, 9, 15)$

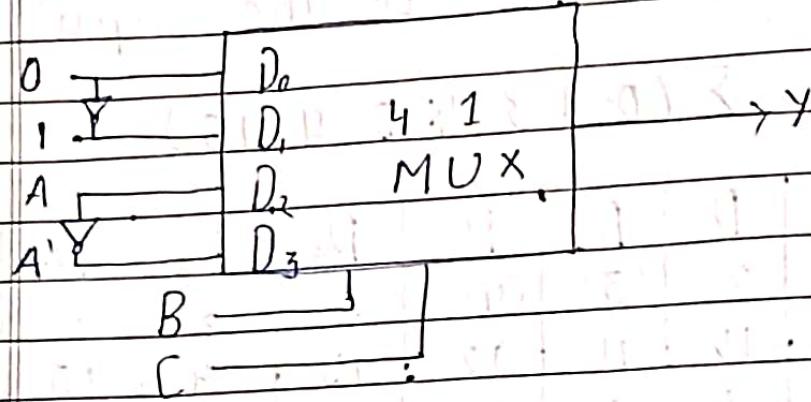
Σ	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
A'	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15



2- $f(A, B, C) = \sum(1, 3, 5, 6)$

$$\underline{A} - (n-1) = \underline{\underline{1}}$$

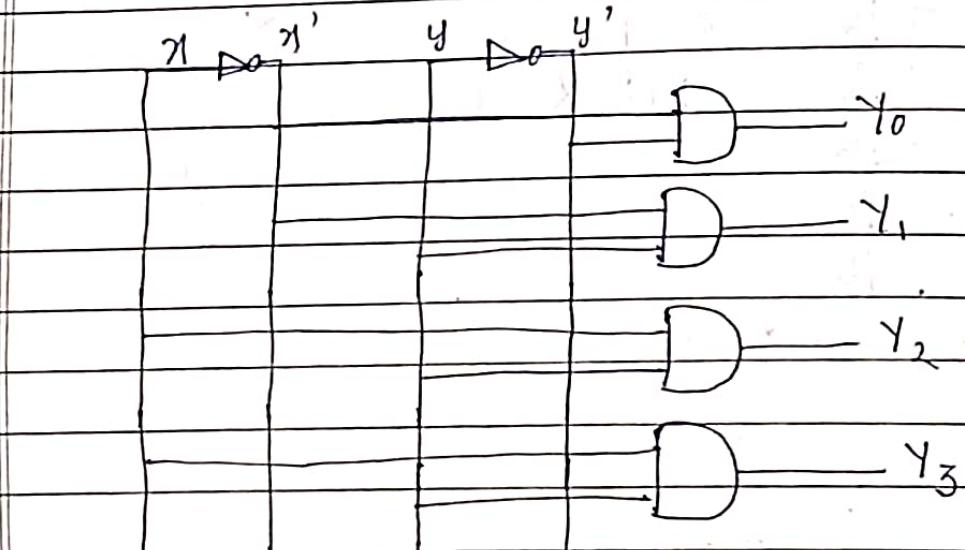
	D_0	D_1	D_2	D_3
A'	0	(1)	1	(3)
A	4	(5)	(6)	7
	0	1	A	A'



Decoders & Encoders

2:4 Decoder

$(n - t_0 - 2^n)$



Give the data in decoded form by the system.

Give The data in encoded form for The
System.
+5Vdc

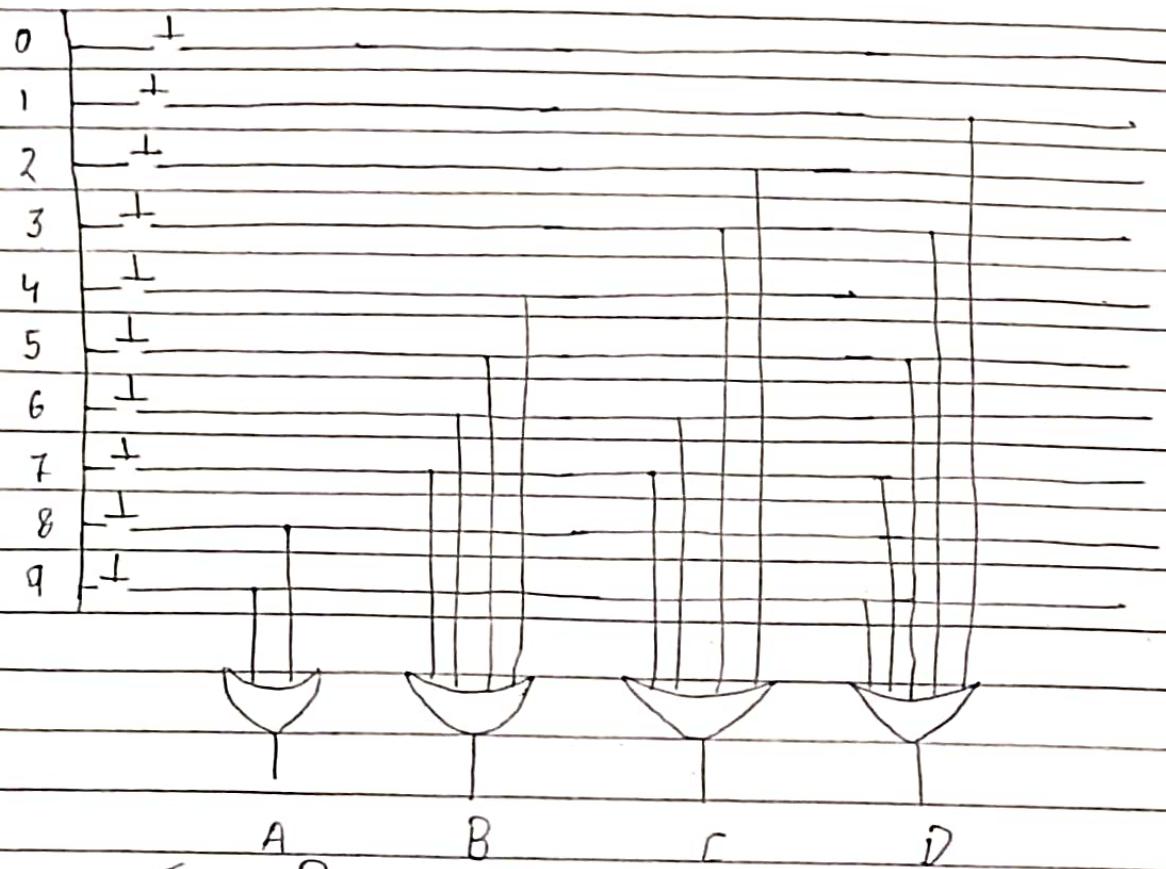
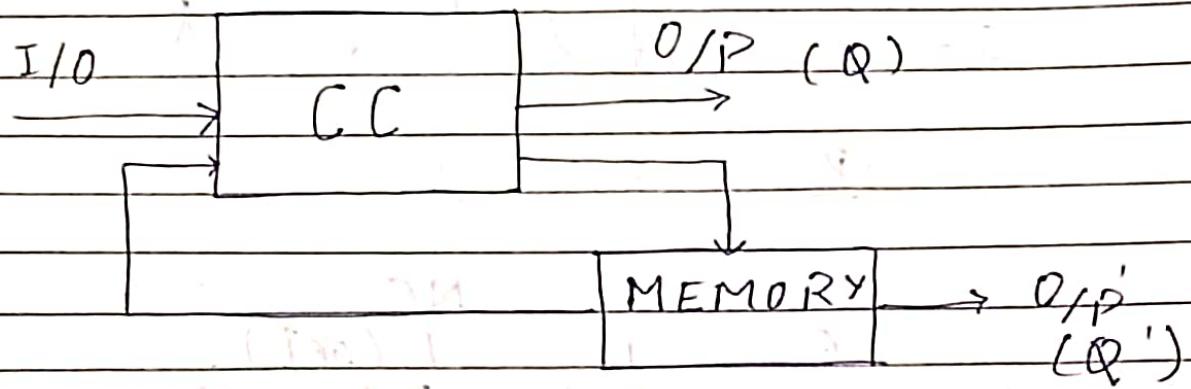


Fig. Decimal to BCD encoder

Design a decimal to BCD converter.

- Q- Design a full adder using nand gates only.
- Q- Design a full subtractor using nore gates only.

Sequential Circuit



Flip-flop

⇒ It is a memory element which can store only one bit of information.

⇒ Main purpose is to store the value temporarily.

1) R-S Flip flop

2) Clocked RS

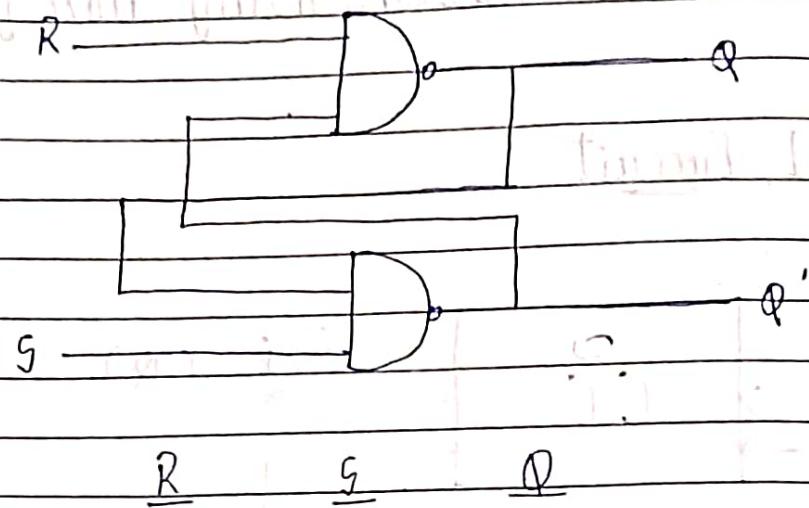
3) Clocked D

4) Clocked JK

5) Clocked T

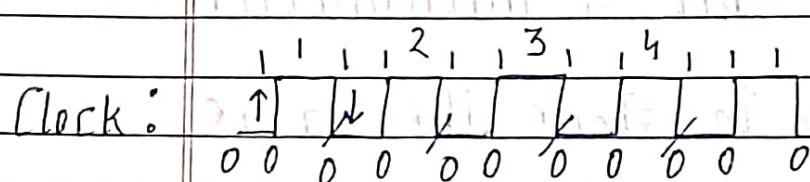
6) Master slave flip flop

1) R-S FlipFlop (R - Reset, S - Set)

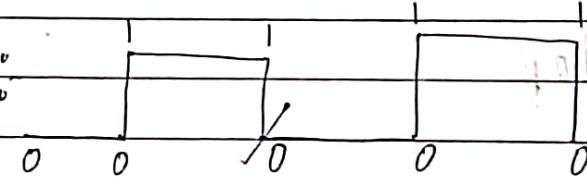


	0	1	NC
0	1	1 (set.)	
1	0	0 (reset)	
1	1	* Race or forbidden cond'	

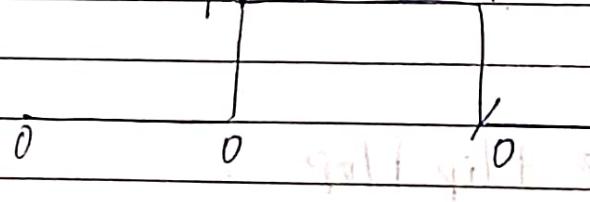
Clock

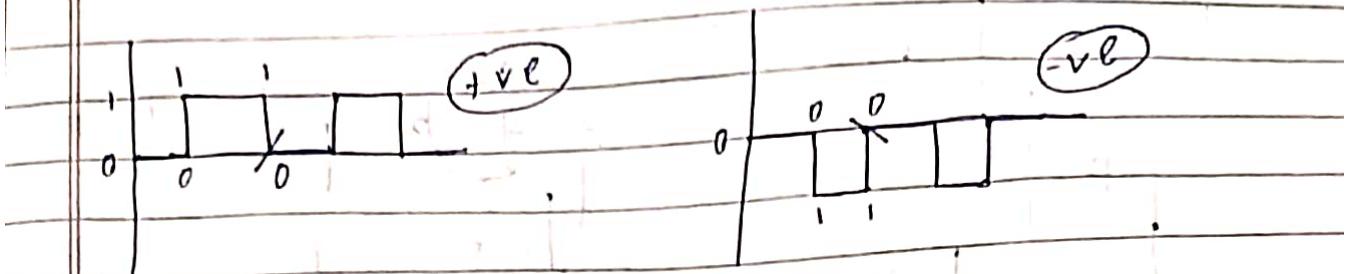


1st FF:

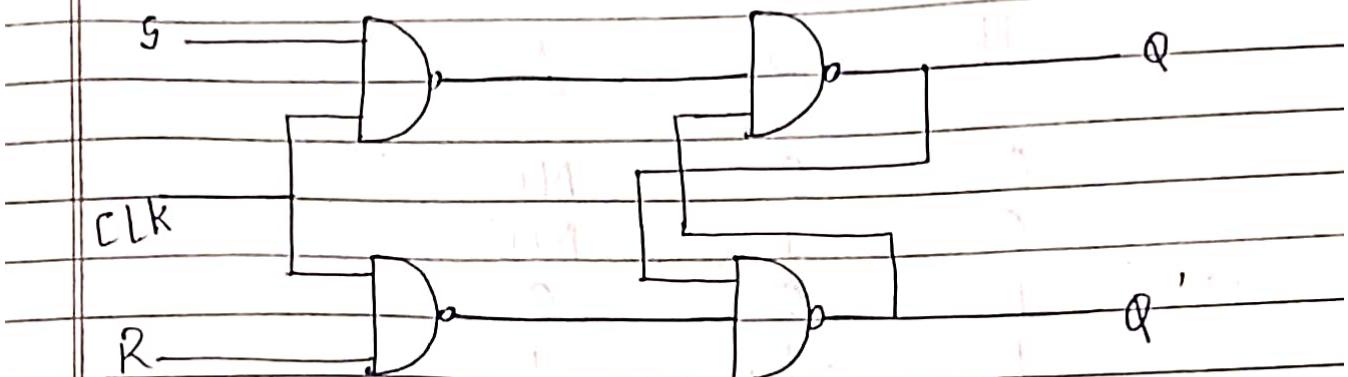


2nd FF:





2) Clocked - RS FlipFlop

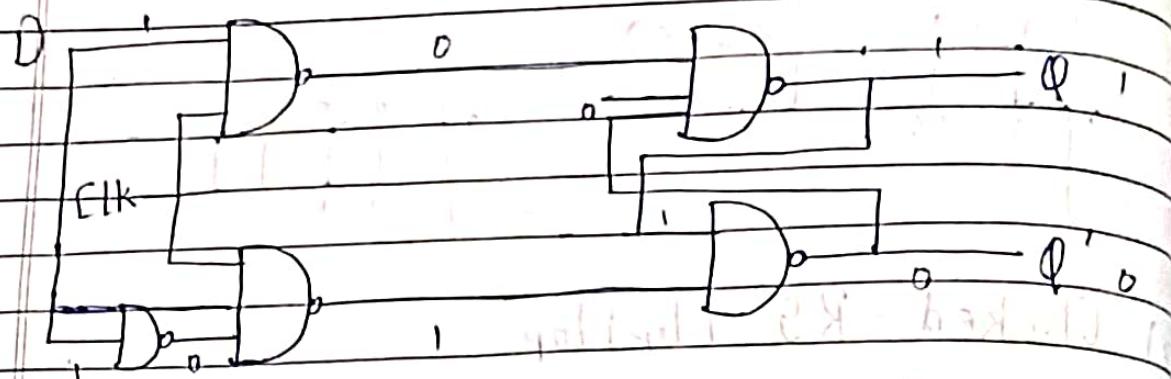


clk R J S with timing and

0	0	0	NC
0	0	1	NC
0	1	0	NC
0	1	1	NC
1	0	0	NC
1	0	1	1 (set)
1	1	0	0 (reset)
1	1	1	* Forbidden or race cond'

3) Clocked - D FlipFlop (D = Data / Delay)

Standard cell
with delay



clk . D . Q

0 0 NC

0 1 NC

1 0 0

1 1 1

Basic Symbol for Flip Flop

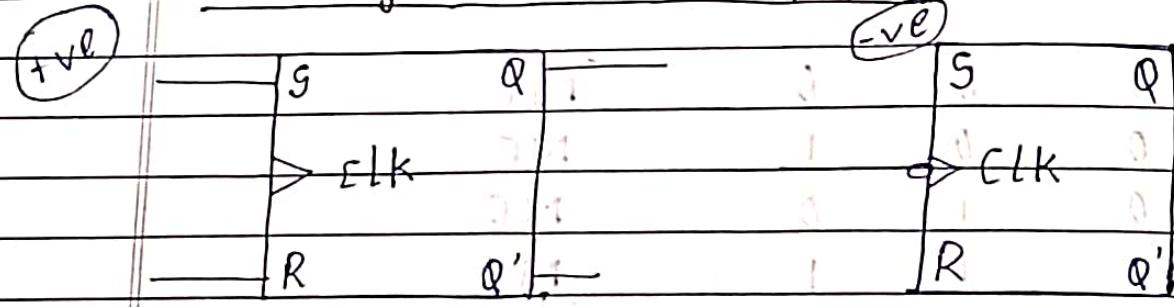


Fig. R-S flip flop (clocked)

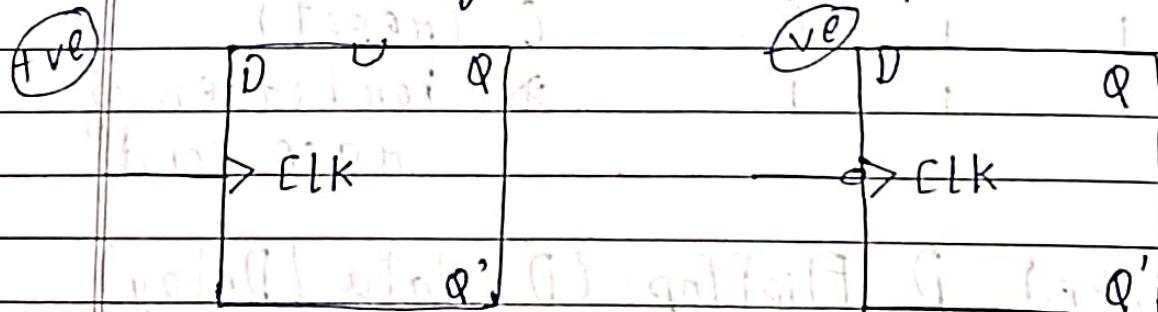
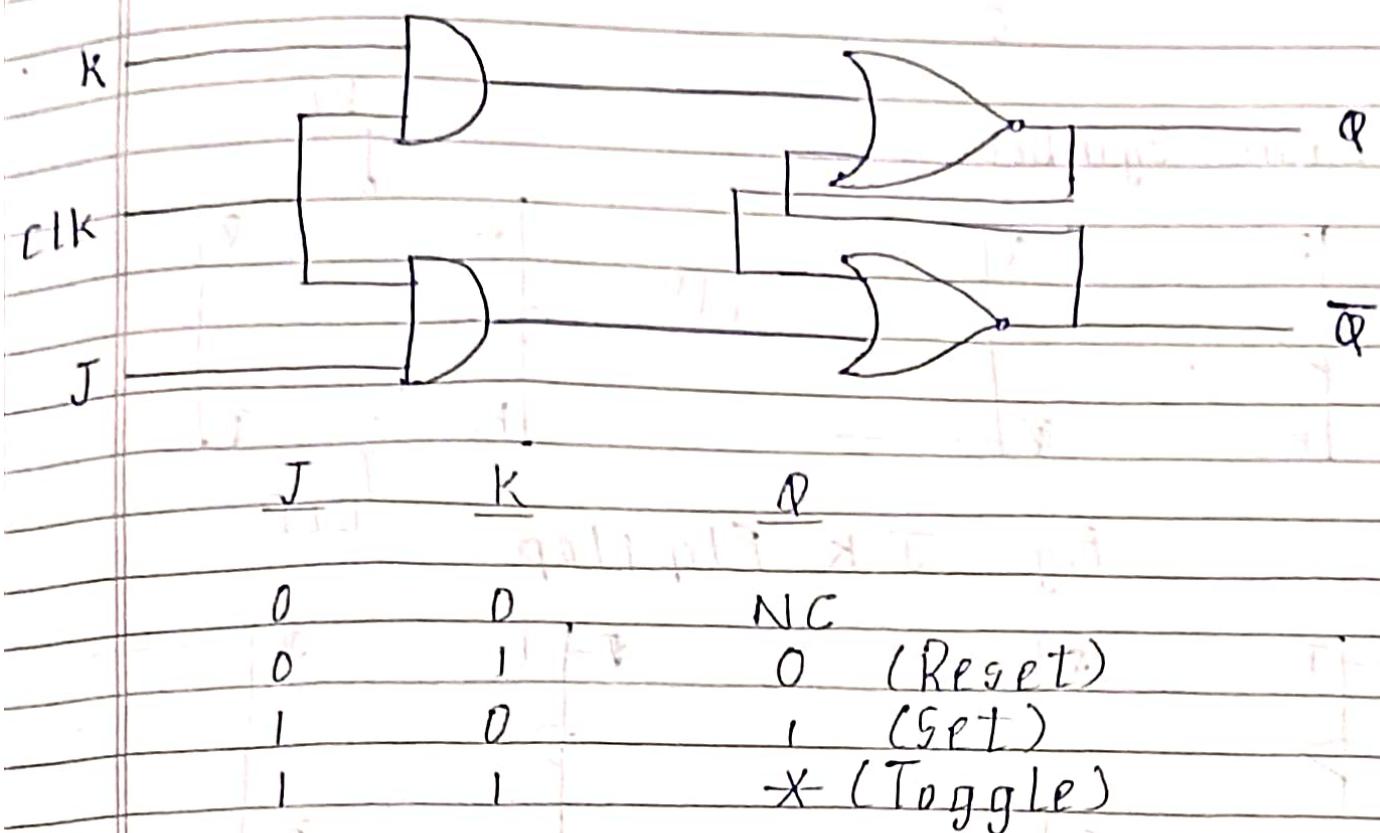
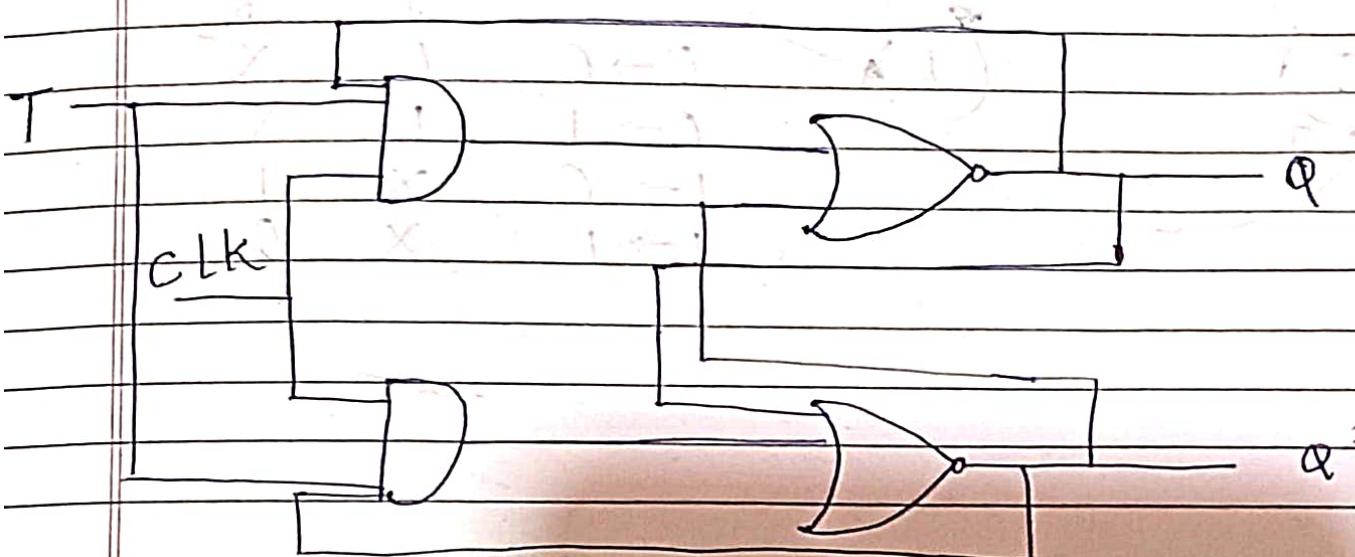


Fig. Clocked D
flip flop

4) Clocked - JK

\Rightarrow Toggle means switch to the opp. state.
 (if it is set condⁿ switch over to
 reset condⁿ or vice-versa).

5) T Flip flop (Toggle)

* Clk → Clear at -ve
 PR → Preset at +ve

Page No. / /

Date: / /

T Q

0 1
1 0

Basic Symbol

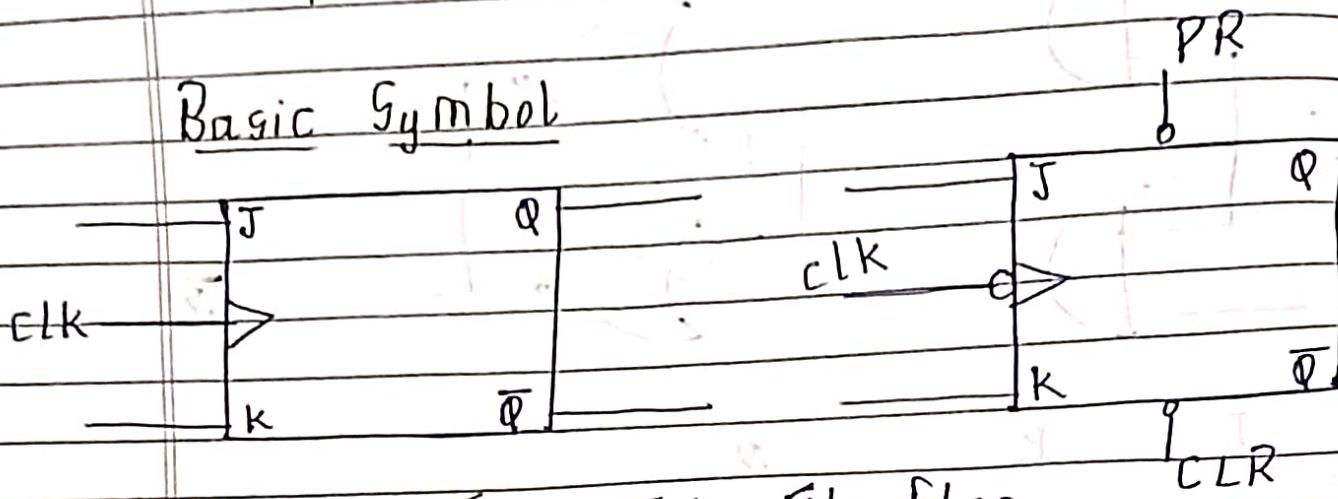


Fig. JK Flipflop

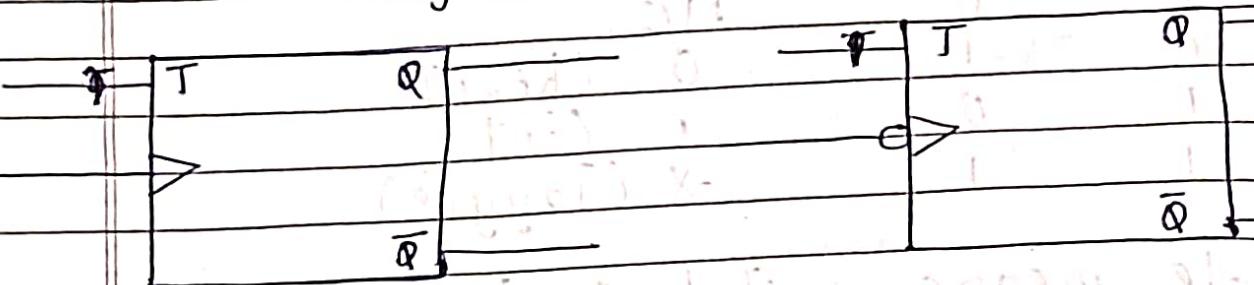


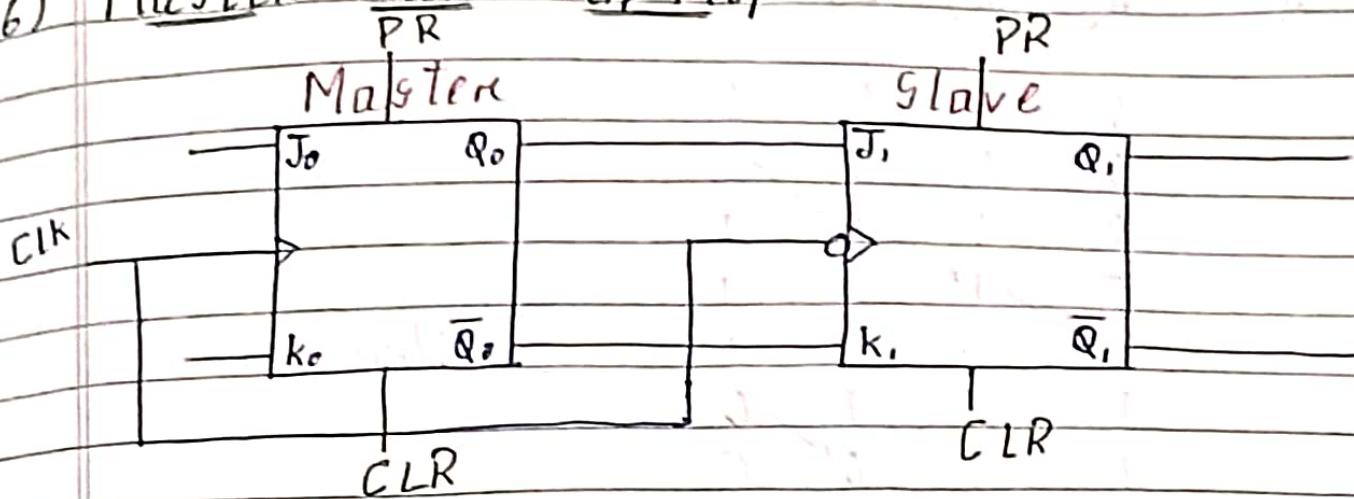
Fig. T FlipFlop

Excitation Table of JK FlipFlop



J	K
0	0
0	1
1	0
1	1

6) Master Slave Flip Flop



~~08/11/24~~ Shift Registers

→ It is a special kind of register which performs the shifting operations.

→ 2 Types :- SHL (Left)
SHR (Right)

SHL

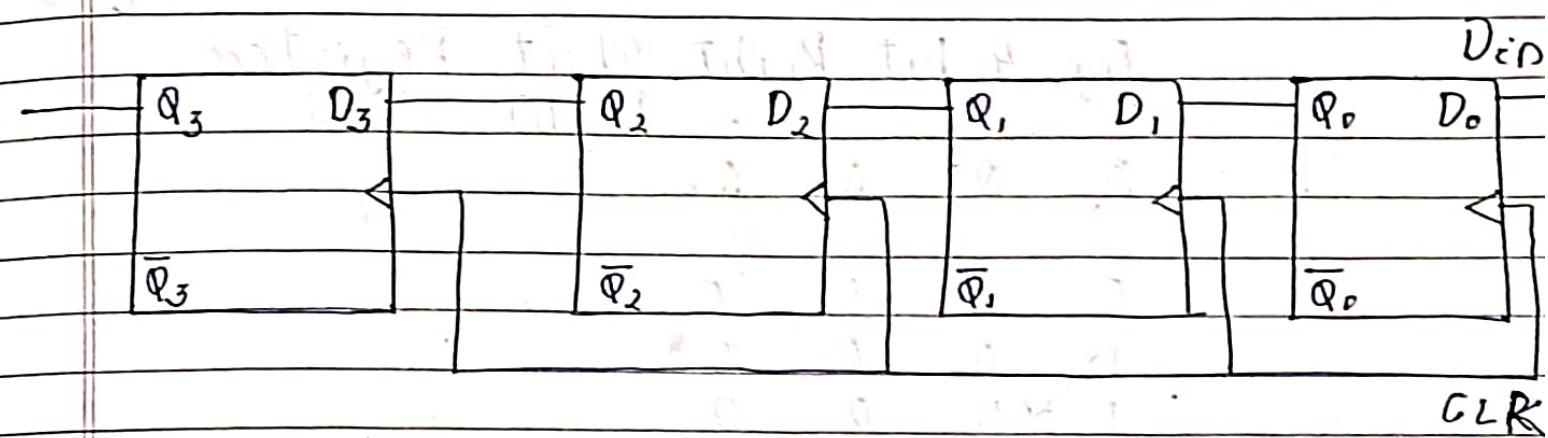
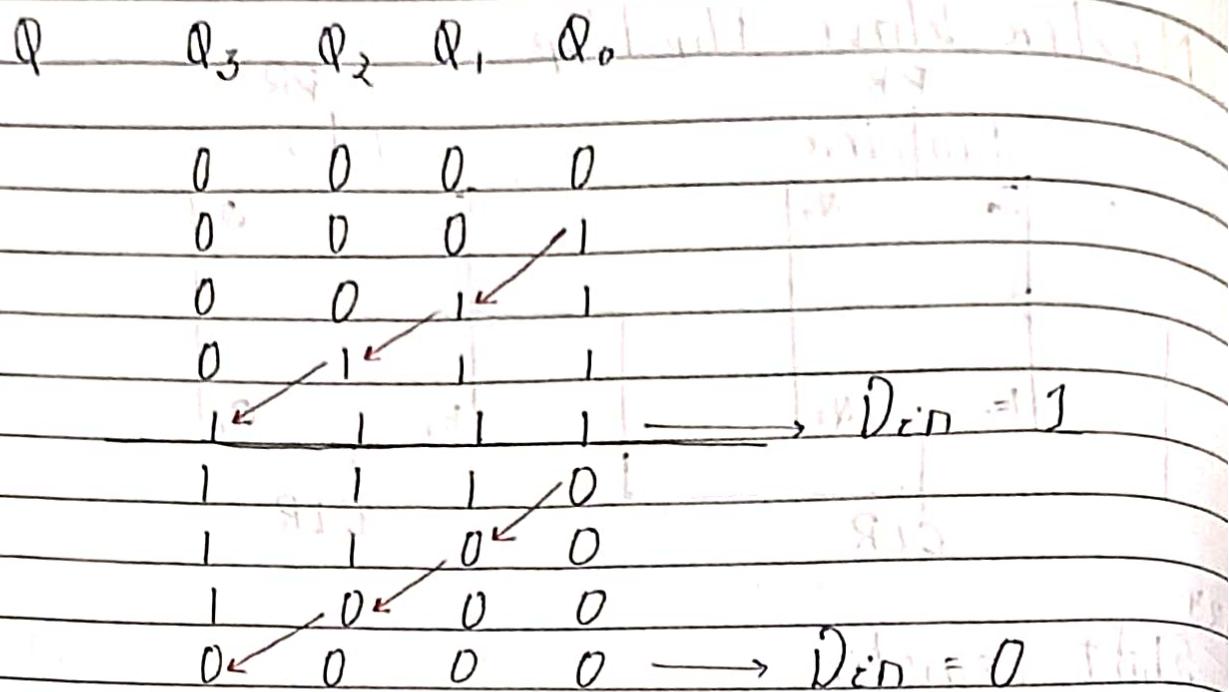


Fig. 4-bit Shift Register

Left shift



SHR

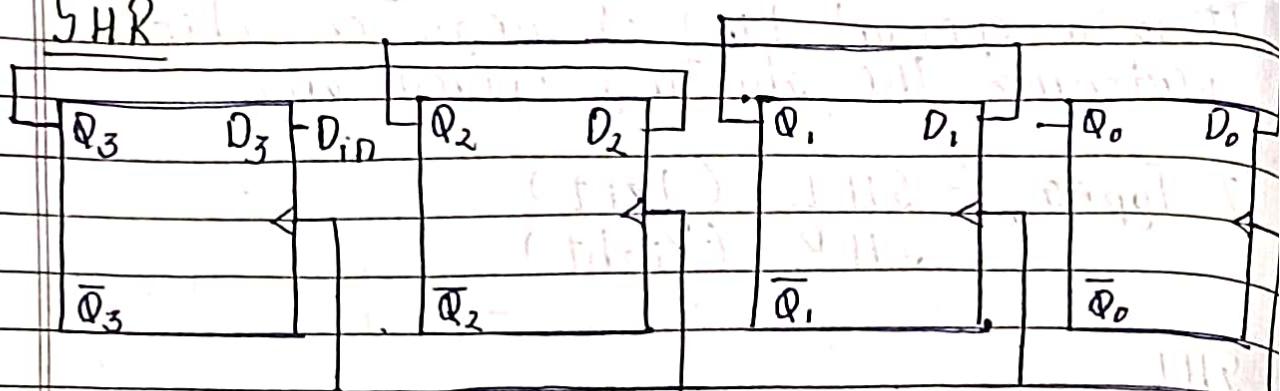
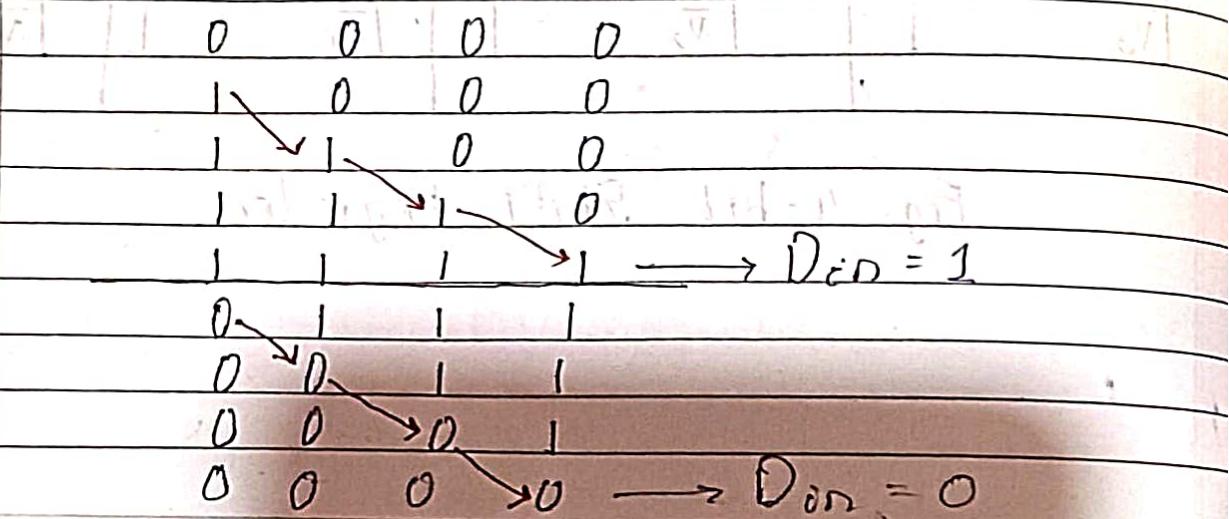


Fig. 4 bit Right Shift Register

$$Q = Q_3 \quad Q_2 \quad Q_1 \quad Q_0$$



Counters

Can perform the counting of clock cycle

2 types :- Asynchronous (Ripple)
Synchronous (Ring)

Ripple Counter

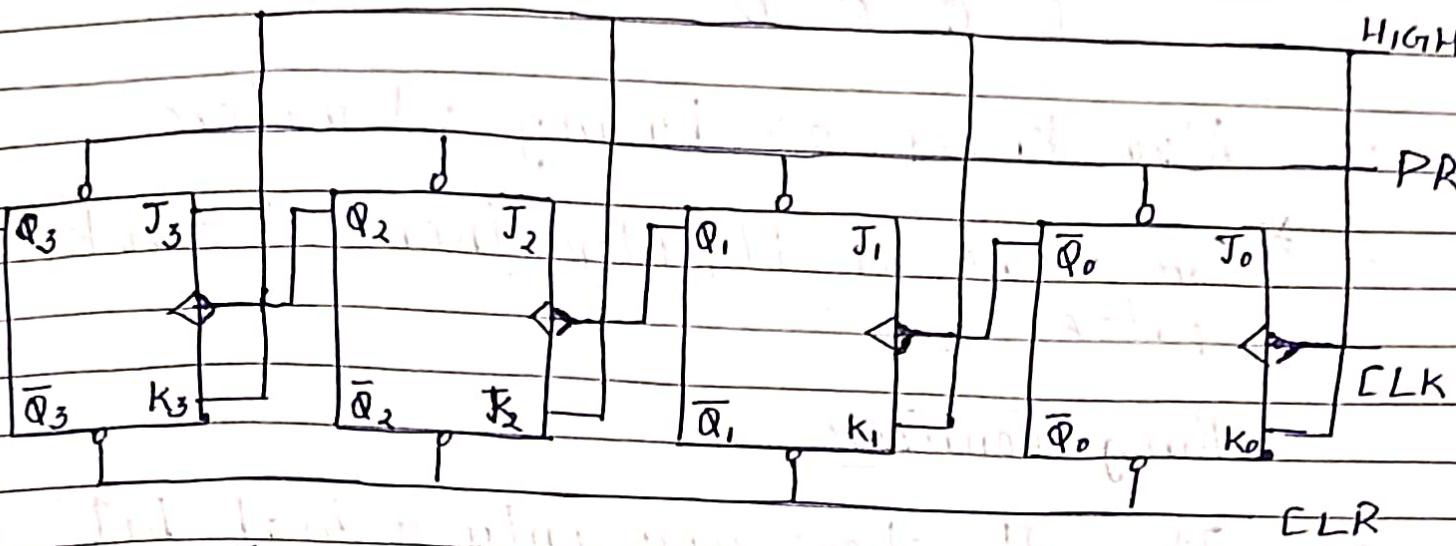


Fig. 4 bit Ripple Counter

$$Q = Q_3 \quad Q_2 \quad Q_1 \quad Q_0$$

0	0	0	0	→	0
0	0	0	1	→	1
0	0	1	0	→	2
0	0	1	1	→	3
0	1	0	0	→	4
0	1	0	1	→	5
0	1	1	0	→	6
0	1	1	1	→	7
1	0	0	0	→	8

$$Q = Q_3 \ Q_2 \ Q_1 \ Q_0$$

$$\begin{array}{cccc} 1 & 0 & 0 & 1 \end{array} \rightarrow 9$$

$$\begin{array}{cccc} 1 & 0 & 1 & 0 \end{array} \rightarrow 10$$

$$\begin{array}{cccc} 1 & 0 & 1 & 1 \end{array} \rightarrow 11$$

$$\begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \rightarrow 12$$

$$\begin{array}{cccc} 1 & 1 & 0 & 1 \end{array} \rightarrow 13$$

$$\begin{array}{cccc} 1 & 1 & 1 & 0 \end{array} \rightarrow 14$$

$$\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array} \rightarrow 15$$

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \end{array}$$

Also known as binary odometer

* For n no. of Flip Flop then 2^n clock pulses

Ring Counter

It will show you only a high bit available at particular word

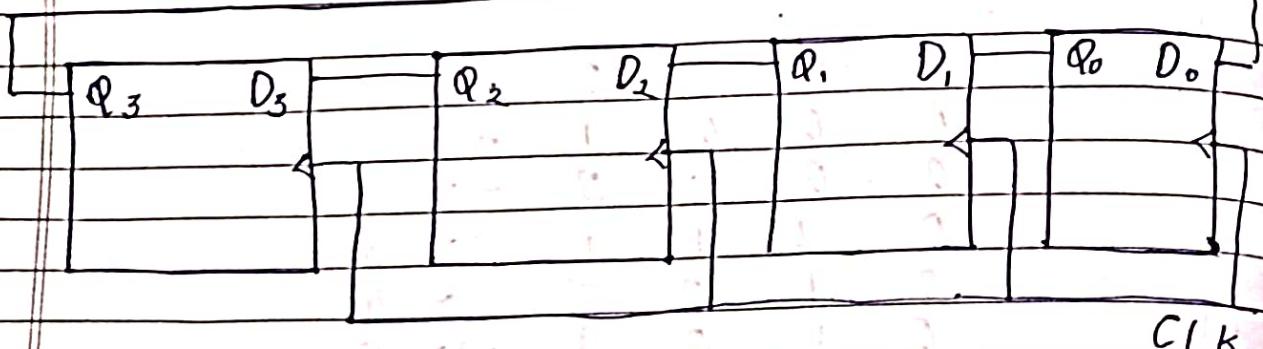


Fig. 4 bit Ring Counter

$$Q = Q_3 \quad Q_2 \quad Q_1 \quad Q_0$$

$$\begin{array}{cccc}
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 \\
 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 0 \\
 1 & 0 & 0 & 0 \\
 \hline
 0 & 0 & 0 & 1
 \end{array}$$

* In synchronous, The clock is attached sequentially to the flip flop while in asynchronous, The clock is attached to the output of 1st flip flop.

~~11/12/24~~
Q- Design a 3 bit counter using JK flip-flops.

A-	<u>Q₂</u>	<u>Q₁</u>	<u>Q₀</u>	FF0 J ₀ K ₀	FF1 J ₁ K ₁	FF2 J ₂ K ₂
0	0	0	0	1 x	0 x	0 x
0	0	1	0	x 1	1 x	0 x
0	1	0	0	1 x	x 0	0 x
0	1	1	0	x 1	x 1	1 x
1	0	0	0	x x	0 x	x 0
1	0	1	0	1 x	1 x	x 0
1	1	0	0	x x	x 0	x 0
1	1	1	0	1 x	x 1	x 1
0	0	0	0			

Q_1, Q_2	00	01	11	10	$J_0 = 1$
0	1	1	1	1	
1	x	x	x	x	

Q_1, Q_2	00	01	11	10	$k_0 = 1$
0	x	x	x	x	
1	1	x	x	1	

Q_1, Q_2	00	01	11	10	$J_1 = Q_0$
0	0	0	x	x	
1	1	1	x	x	

Q_1, Q_2	00	01	11	10	$k_1 = Q_0$
0	x	x	0	0	
1	x	x	1	1	

Q_1, Q_2	00	01	11	10	$J_2 = Q_0, Q_1$
0	0	x	x	0	
1	0	x	x	1	

Q_1, Q_2	00	01	11	10	$k_2 = Q_0, Q_1$
0	x	0	0	x	
1	x	0	1	x	

$$\begin{aligned}
 J_0 &= K_0 = 1 \\
 J_1 &= K_1 = Q_1 \\
 J_2 &= K_2 = Q_0 Q_1
 \end{aligned}$$

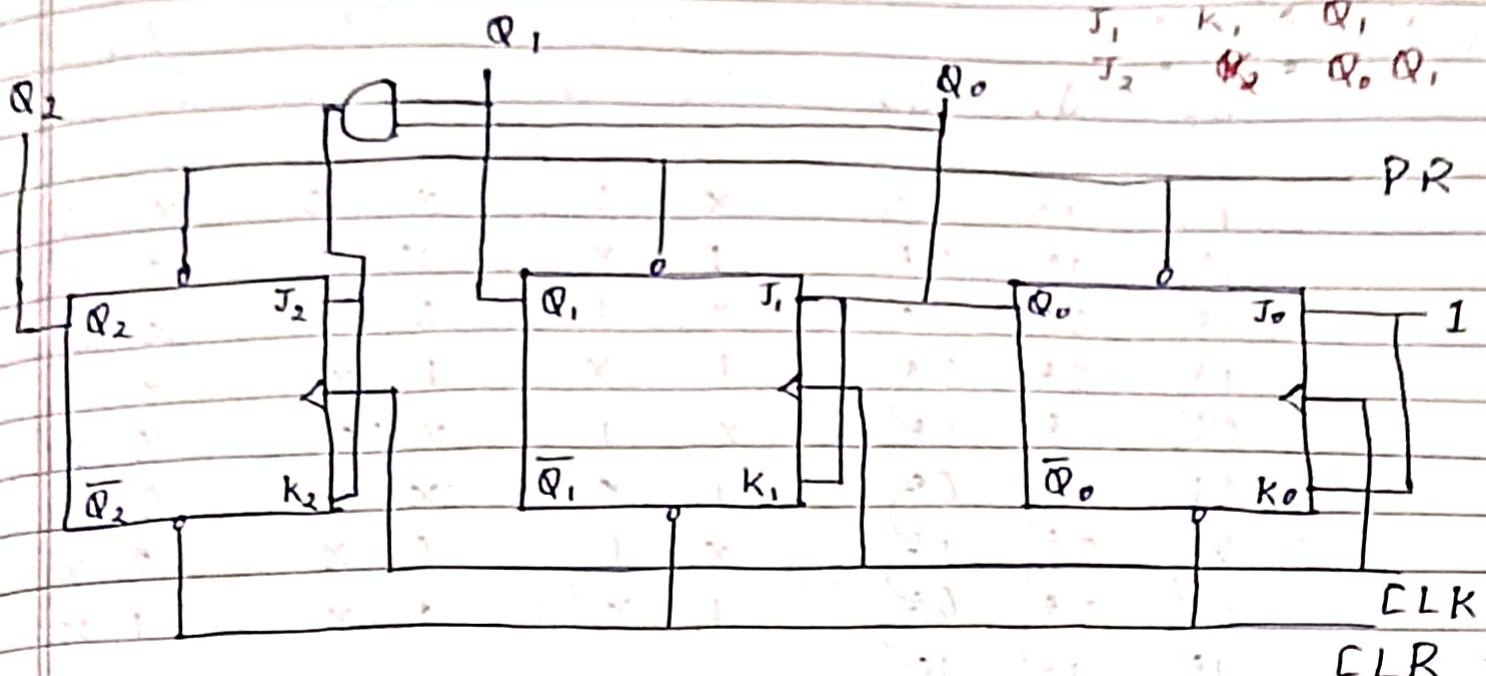


Fig. 3 bit Counter (MOD-8 COUNTER)
(3 bit Up Counter)

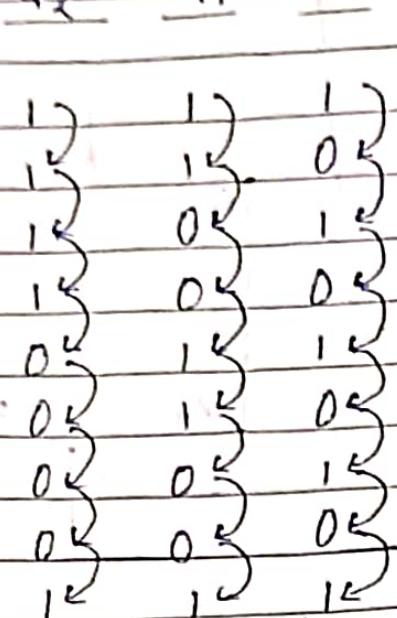
- q- Design a 3 bit Down Counter.
- * q- Design a 3 bit Up / Down Counter.
- * q- Design a 4 bit Up / Down Counter.
- q- Design a decade counter. (Decimal counter)
(Mod-10 Counter)

ANSWERS

~~3-bit Down Counter~~

$Q_2 \quad Q_1 \quad Q_0$

FF0 : $J_0 \quad K_0$; FF1 : $J_1 \quad K_1$; FF2 : $J_2 \quad K_2$



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

Q_0	Q_1	Q_2	00	01	11	10
0	1	1	1	1		
1	x	x	x	x		

$$J_0 = 1$$

Q_0	Q_1	Q_2	00	01	11	10
0	x	x	x	x		
1	1	1	1	1		

$$K_0 = 1$$

Q_0	Q_1	Q_2	00	01	11	10
0	1	1	x	x		
1	0	0	x	x		

$$\text{with } J_1 = Q_0' - G_1 T_1$$

Q_0	Q_1	Q_2	00	01	11	10
0	x	x	1	1		
1	x	x	0	0		

$$k_1 = Q_0'$$

Q_2, Q_1	00	01	11	10
0	1	X	X	0
1	0	X	X	0

$$J_2 = Q_0' Q_1'$$

Q_2, Q_1	00	01	11	10
0	X	1	0	X
1	X	0	0	X

$$k_2 = Q_0' Q_1'$$

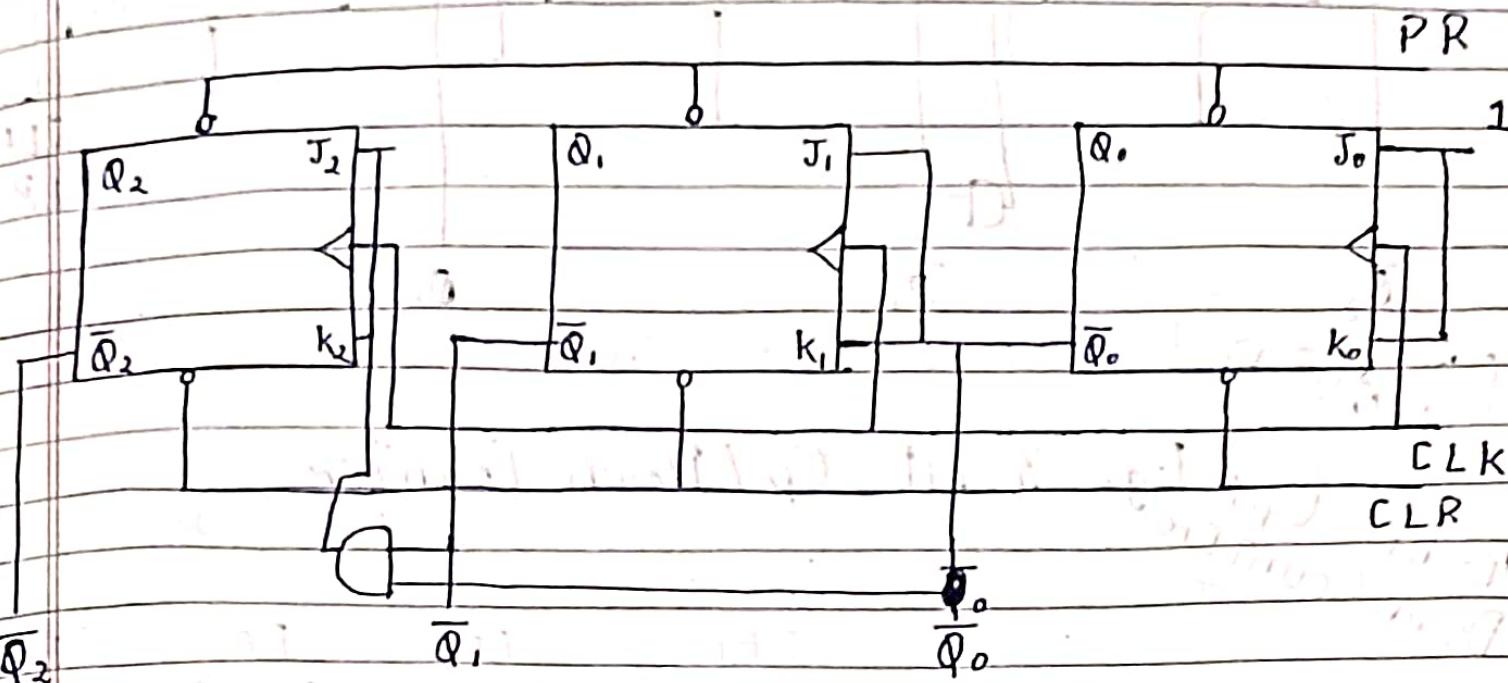
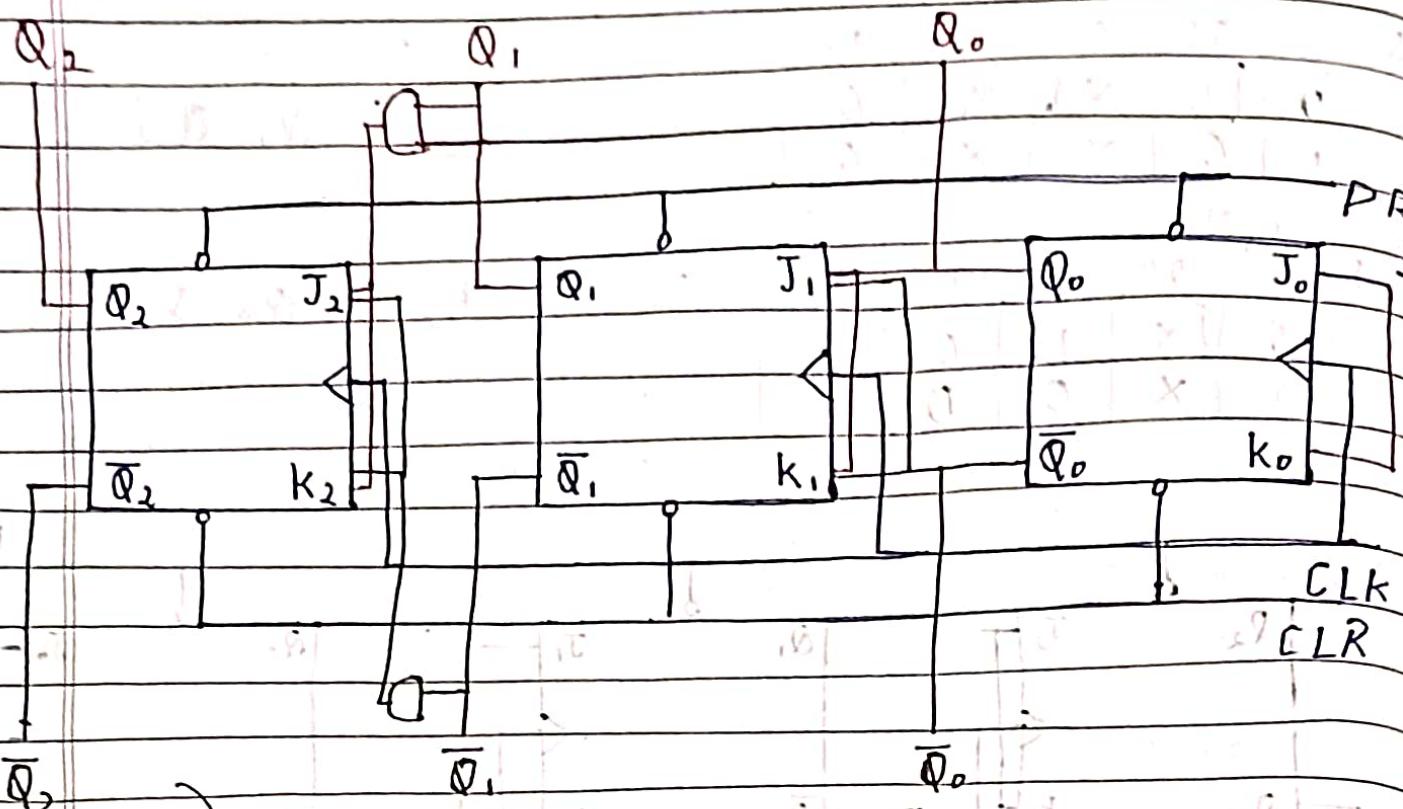


Fig. 3 bit Down Counter

3 Bit Up / Down Counter

- STEP 1 :- Fig. of Up Counter
- STEP 2 :- Fig. of Down Counter
- STEP 3 :- Fig. of Comb " fig. of Up / Down Counter "



(Down)

Fig. 3 bit Up/Down Counter

4-bit Up
Down Counter

Q_3	Q_2	Q_1	Q_0	J_0	K_0	J_1	K_1	J_2	K_2	J_3	K_3
0	0	0	0	1	x	0	x	0	x	0	x
0	0	0	1	x	1	x	0	x	0	x	0
0	0	1	0	1	x	0	0	0	x	0	x
0	0	1	1	x	1	x	1	x	x	x	0
0	1	0	0	1	x	0	x	x	0	0	x
0	1	0	1	x	1	x	1	x	0	0	x
0	1	1	0	1	x	0	0	x	0	0	x
0	1	1	1	x	1	x	x	x	1	1	x
1	0	0	0	1	x	0	x	0	x	x	0
1	0	0	1	x	1	x	0	x	0	x	*
1	0	1	0	x	1	1	x	0	y	x	0
1	0	1	1	1	x	0	0	x	y	x	0

Q_3	Q_2	Q_1	Q_0	J_0	k_0	J_1	k_1	J_2	k_2	J_3	k_3
0	0	1	1	x	1	x	1	1	x	x	0
1	0	0	0	1	x	0	x	x	0	x	0
1	1	0	1	x	1	1	x	x	0	x	0
1	1	1	0	1	x	x	0	x	0	x	0
1	1	1	1	x	1	x	1	x	1	x	1
0	0	0	0								

Q_3	Q_2	Q_1	Q_0	00	01	11	10
Q_0	Q_1	Q_2	Q_3	00	01	11	10
00	1	1	1	1	1	1	1
01	1	1	1	1	1	1	1
11	x	x	x	x	x	x	x
10	x	x	x	x	x	x	x

$$J_0 = 1$$

Q_3	Q_2	Q_1	Q_0	00	01	11	10
Q_0	Q_1	Q_2	Q_3	00	01	11	10
00	x	x	x	x	x	x	x
01	x	x	x	x	x	x	x
11	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1

$$k_0 = 1$$

Q_3	Q_2	Q_1	Q_0	00	01	11	10
Q_0	Q_1	Q_2	Q_3	00	01	11	10
00	0	0	0	0	0	0	0
01	x	x	x	x	x	x	x
11	x	x	x	x	x	x	x
10	01	1	1	1	1	1	1

$$J_1 = Q_0$$

Q_3	Q_2	Q_1	Q_0	00	01	11	10
Q_0	Q_1	Q_2	Q_3	00	01	11	10
00	x	x	x	x	x	x	x
01	0	0	0	0	0	0	0
11	1	1	1	1	1	1	1
10	x	x	x	x	x	x	x

$$k_1 = Q_0$$

$Q_0 Q_1$	$Q_2 Q_3$	00	01	11	10
00	00	0	X	X	X
01	00	0	X	X	X
11	11	1	X	X	X
10	00	0	X	X	X

$$J_2 = Q_0 Q_1 \text{ (Q2)}$$

$Q_0 Q_1$	$Q_2 Q_3$	00	01	11	10
00	X	X	0	0	0
01	X	X	0	0	0
11	X	X	1	1	1
10	X	X	0	0	0

$$K_2 = Q_0 Q_1 \text{ (Q2)}$$

$Q_0 Q_1$	$Q_2 Q_3$	00	01	11	10
00	0	0	X	0	0
01	0	0	X	X	X
11	X	X	X	1	1
10	0	X	X	0	0

$$J_3 = Q_0 Q_1 Q_2 \text{ (Q3)}$$

$Q_0 Q_1$	$Q_2 Q_3$	00	01	11	10
00	X	0	0	X	X
01	X	0	0	X	X
11	X	0	1	X	X
10	X	0	0	X	X

$$K_3 = Q_0 Q_1 Q_2 \text{ (Q3)}$$

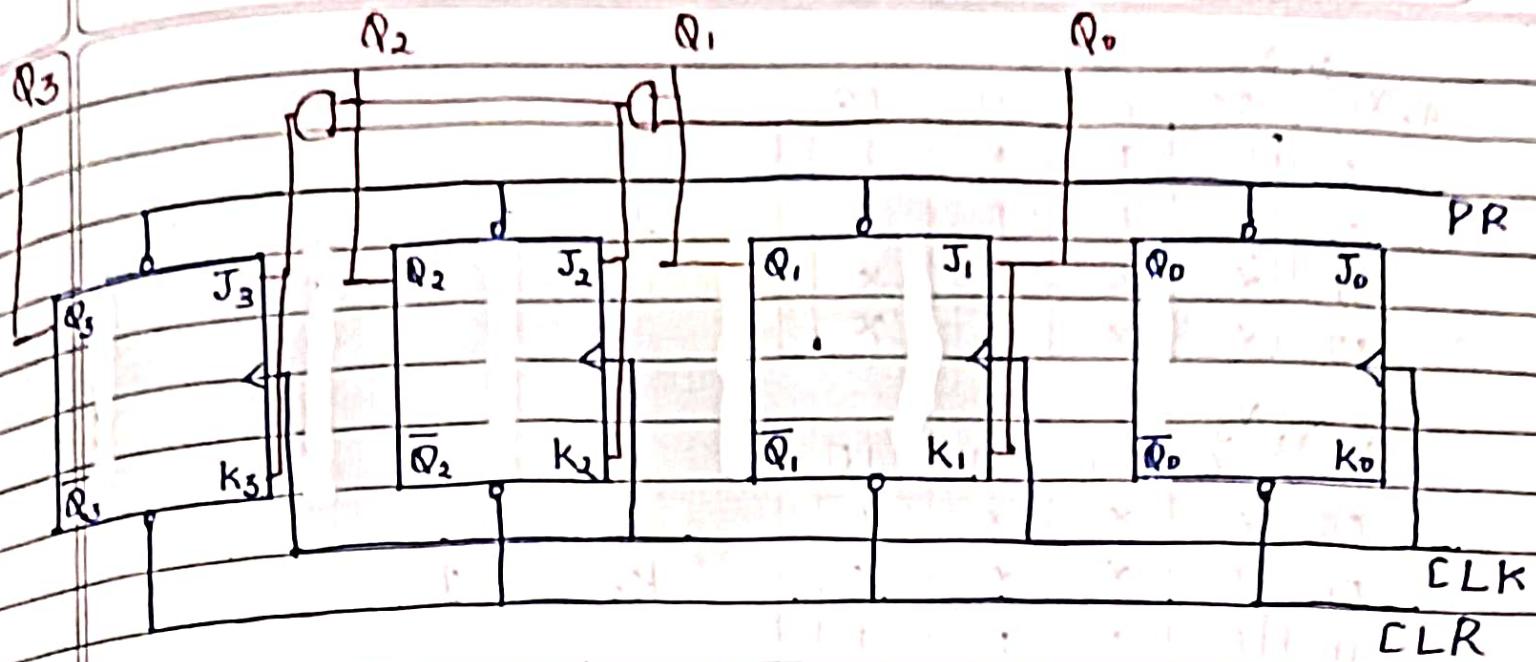


Fig. 4 bit Up Counter

<u>Q_3</u>	<u>Q_2</u>	<u>Q_1</u>	<u>Q_0</u>	<u>J_0</u>	<u>K_0</u>	<u>J_1</u>	<u>K_1</u>	<u>J_2</u>	<u>K_2</u>	<u>J_3</u>	<u>K_3</u>
1	1	1	1	X	1	-X	0	X	0	X	0
1	1	1	0	1	X	X	1	X	0	X	0
1	1	0	1	X	1	0	X	X	0	X	0
1	1	0	0	1	X	1	X	X	1	X	0
1	0	1	1	X	1	X	0	1	0	X	0
1	0	1	0	1	X	1	X	0	X	X	0
1	0	0	1	-X	1	0	X	0	X	X	0
1	0	0	0	1	X	1	X	1	X	X	1
0	1	1	1	X	1	X	0	X	1	0	X
0	1	1	0	1	X	X	1	X	0	0	X
0	1	0	1	X	1	0	X	X	0	0	X
0	1	0	0	1	X	1	X	X	1	0	X
0	0	1	1	X	1	X	0	X	0	0	X
0	0	1	0	1	X	X	1	0	X	0	X
0	0	0	1	X	1	0	X	0	X	0	X
0	0	0	0	1	X	1	X	1	X	1	X

Q_0, Q_1	Q_2, Q_3	00	01	11	10
00		1	1	1	1
01		1	1	1	1
11		x	x	x	x
10		x	x	x	x

$$J_0 = 1$$

Q_0, Q_1	Q_2, Q_3	00	01	11	10
00		x	x	x	x
01		x	x	x	x
11		1	1	1	1
10		1	1	1	1

$$k_0 = 1$$

Q_0, Q_1	Q_2, Q_3	00	01	11	10
00		1	1	1	1
01		x	x	x	x
11		x	x	x	x
10		0	0	0	0

$$J_{12} = Q_0' Q_3$$

Q_0, Q_1	Q_2, Q_3	00	01	11	10
00		x	x	x	x
01		1	1	1	1
11		0	0	0	0
10		x	x	x	x

$$k_1 = Q_0'$$

Q_0, Q_1	Q_2, Q_3	00	01	11	10
00		1	1	x	x
01		0	0	x	x
11		0	0	x	x
10		0	0	x	x

$$J_2 = Q_0' Q_1$$

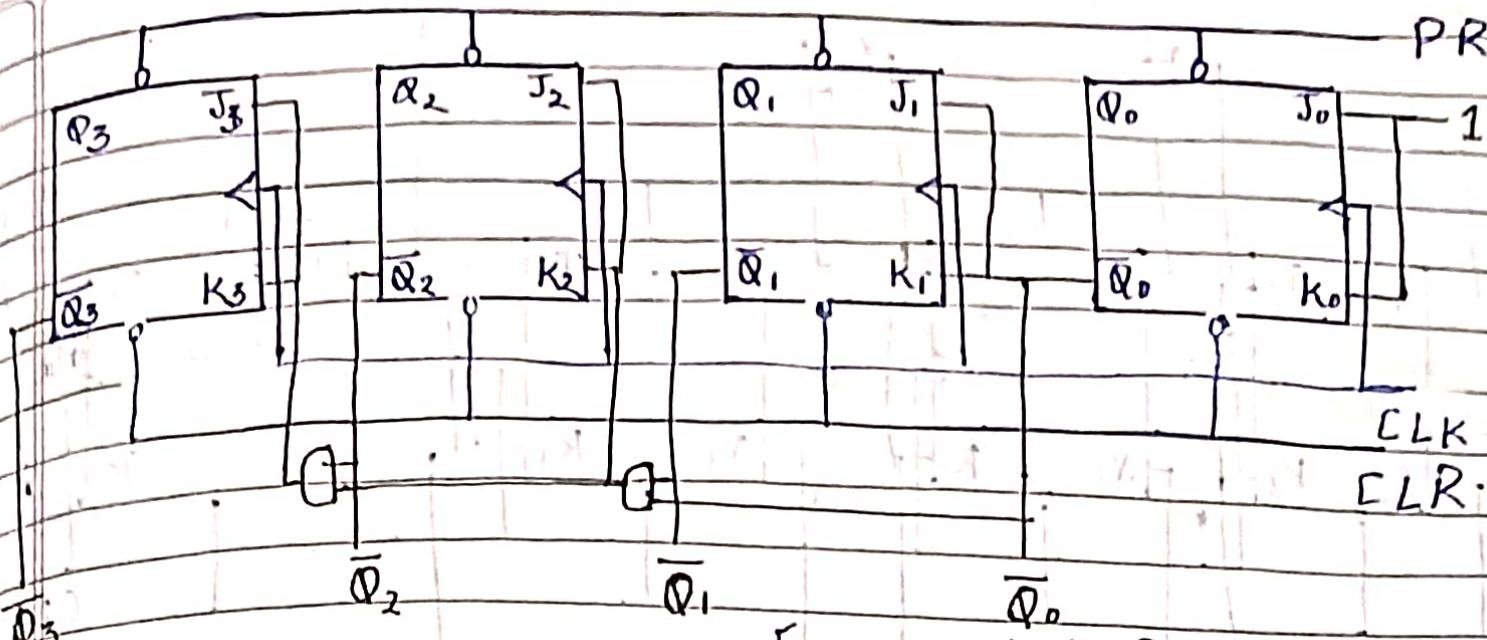


Fig. 4 bit Down Counter

	$Q_2 Q_3$	00	01	11	10
$Q_0 Q_1$	X	X	1	1	
00	X	X	0	0	
01	X	X	0	0	
11	X	X	0	0	
10	X	X	0	0	

$$K_2 = Q_0' Q_1' Q_2'$$

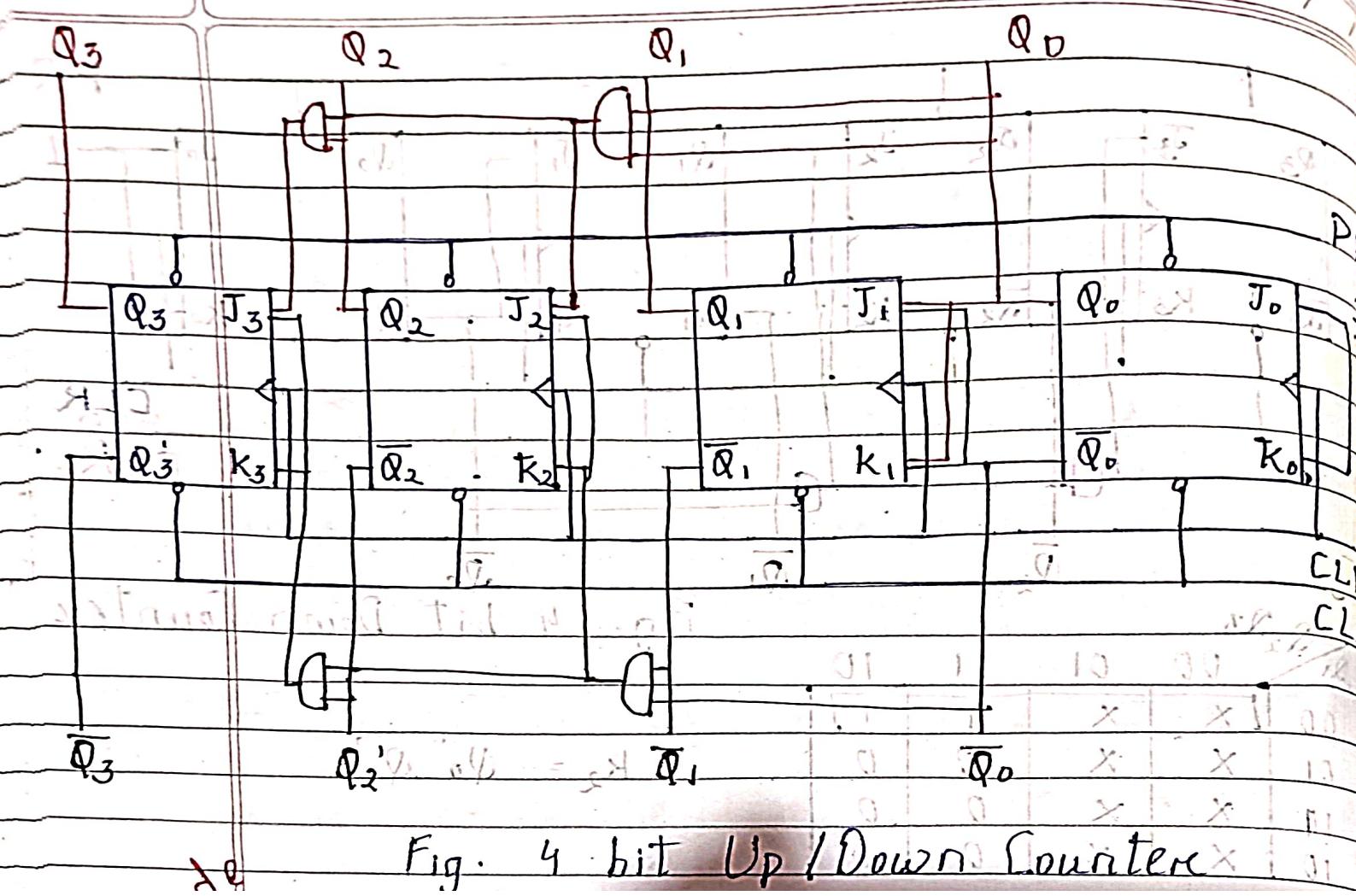
	$Q_2 Q_3$	00	01	11	10
$Q_0 Q_1$	1	X	X	0	
00	0	X	X	0	
01	0	X	X	0	
11	0	X	X	0	
10	0	X	X	0	

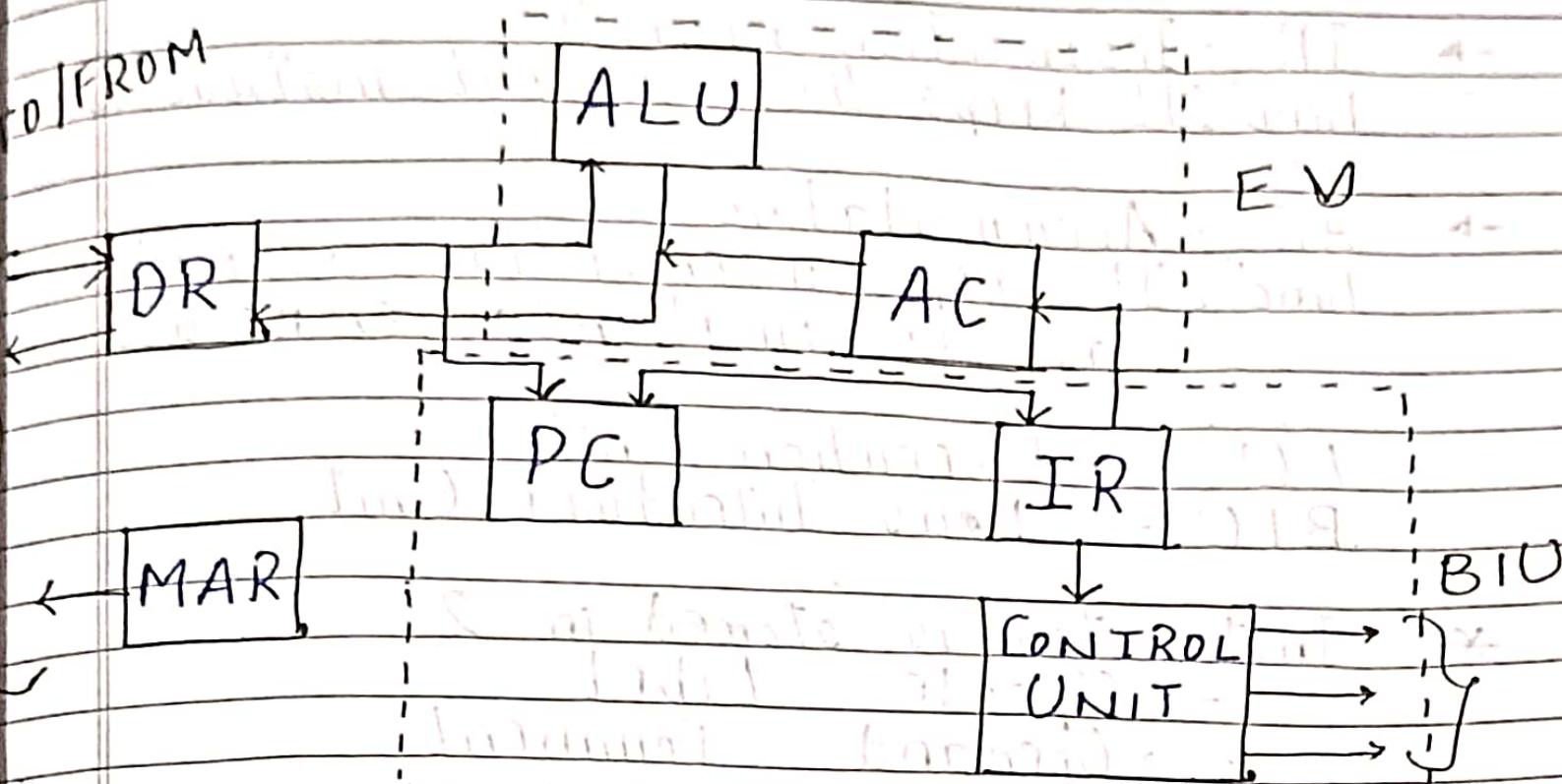
$$J_3 = Q_0' Q_1' Q_2'$$

	$Q_2 Q_3$	00	01	11	10
$Q_0 Q_1$	X	1	0	X	
00	X	0	0	X	
01	X	0	0	X	
11	X	0	0	X	
10	X	0	0	X	

$$K_3 = Q_0' Q_1' Q_2'$$

Page No. /
Date: /



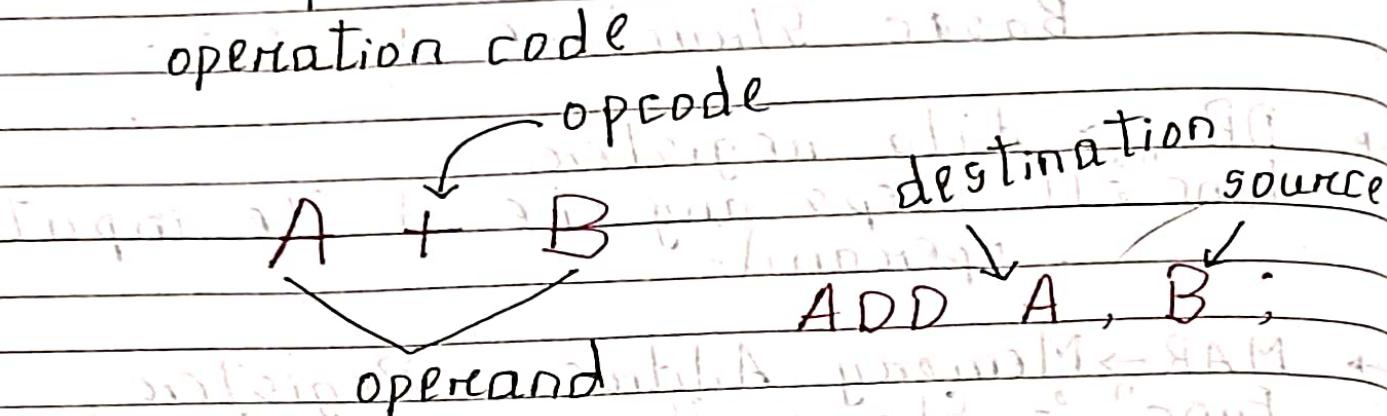


Von Neumann Architecture Or Basic Structure of CPU

- DR → data register.
Func :- It keeps any one of the input operand.
- MAR → Memory Address Register
Func :- Shows the memory location where the data will be stored and vice-versa.
- PC → Program Counter
It keeps the next instruction to be executed.

- IR → Instruction register
Func: It keeps the current instruction
- AC → Accumulator
Func: It keeps intermediate as well as the final register result.
- * Instruction is stored in 2
 - OPCODE Label
 - Operand Comment

Label : \uparrow OPCODE Operand(s); Comment



AddresSES of Instruction

5 types :- 3 address / bytes

2 address / bytes

1 address / bytes

0 address / bytes

(R) → Register

$$P: \quad X = (A + B * (C + D))$$

3 by 2s
 ADD R₁, A, B : R₁ ← M[A] + M[B]
 ADD R₂, C, D : R₂ ← M[C] + M[D]
 MUL X, R₁, R₂ : M[X] ← R₁ * R₂.

2 address Instruction (MOV)

MOV R ₁ , A :	R ₁ ← M[A]
ADD R ₁ , B :	R ₁ ← R ₁ + M[B]
MOV R ₂ , C :	R ₂ ← M[C]
ADD R ₂ , D :	R ₂ ← R ₂ + M[D]
MUL R ₁ , R ₂ :	R ₁ ← R ₁ * R ₂
MOV X, R ₁ :	M[X] ← R ₁

→ 2 operand instruction
 → special operators MOV

1 address Instruction

- A register called accumulator is hidden.
- 2 hidden instruction → LOAD
STORE

- * Store from accumulators
- * Load to accumulators

Page No. _____

Date: / /

LOAD A ; $AC \leftarrow M[A]$
 ADD B ; $AC \leftarrow AC + M[B]$
 STORE T ; $M[T] \leftarrow AC$
 LOAD C ; $AC \leftarrow M[C]$
 ADD D ; $AC \leftarrow AC + M[D]$
 MULTI T ; $AC \leftarrow AC * M[T]$
 STORE X ; $M[X] \leftarrow AC$

Code equivalent assembly

$R1 \leftarrow M[A]$
 $R2 \leftarrow M[B]$
 $R3 \leftarrow R1 + R2$
 $R4 \leftarrow M[C]$
 $R5 \leftarrow R4 + R3$
 $R6 \leftarrow M[D]$
 $R7 \leftarrow R5 + R6$
 $R8 \leftarrow R7 * R6$
 $R9 \leftarrow EXIT$

Assembly language ←
→ VM language ←
→ high-level language ←

mid-level language ←

ei abstrahieren, fallen mit dem A
assemblieren

→ Quell - Anweisungen mit dem S
assemblieren

* Store from accumulator
* Load to accumulator

Page No. _____

Date: / /

LOAD A ; $AC \leftarrow M[A]$

ADD B ; $AC \leftarrow AC + M[B]$

STORE T ; $M[T] \leftarrow AC$

LOAD C ; $AC \leftarrow M[C]$

ADD D ; $AC \leftarrow AC + M[D]$

MUL T ; $AC \leftarrow AC * M[T]$

STORE X ; $M[X] \leftarrow AC$

19/11/24

0 address Instruction

→ Stack pointer is a special kind of register which always points to the top of the stack.

PUSH POP



Underflow

PUSH A B → A+B

PUSH B C → A+B+C

ADD D → A+B+C+D

PUSH C D → A+B+C+D+E

ADD E → A+B+C+D+E+F

MUL F → A+B+C+D+E+F*G

POP X

Registers Organisation

Programmable
Visible
Registers

Status OR
cond" code OR
Flag Registers

4 types of Micro operation

- 1) Register Transfer
- 2) Arithmetic Transfer
- 3) Logic Transfer
- 4) Shift Transfer

1) Register Transfer Micro Operation

R M

R R'

R M

M R

M M (Cannot be done directly)

MOV R₁, R₂ (R to R)

MOV R₁, A (R to M)

MOV A, R₁ (M to R)

MOV R₁, A } M to M
 MOV B, R₁ }

2) Arithmetic Transfer Micro Operation

ADD R₁, R₂; R₁ \leftarrow R₁ + R₂

ADD R₁, A; R₁ \leftarrow R₁ + M[A]

ADD A, R₁; M[A] \leftarrow M[A] + R₁

$$R_1 = R_1 + R_2$$

$$\begin{aligned} R_1 &= R_2 + R_3 \\ &= R_2 + (R_3 + 1) \end{aligned}$$

$$R_1 \leftarrow R_1 + 1$$

$$R_1 \rightarrow R_1 - 1$$

3) Logic Transfer Micro Operation

AND R_1, R_2

$$R_1 = 1011 \text{ (Vanishes)}$$

$$R_2 = 0101$$

$$R_1 = 0001$$

$$R_1 \leftarrow R_1 \wedge R_2$$

4) Shift Transfer Micro Operation

SHL } LOGICAL

SHR }

ASHL } ARITHMETIC

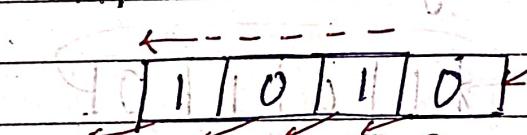
ASHR }

CIL } CIRCULAR

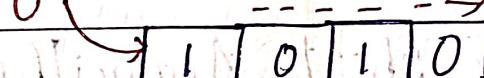
CIR }

SHL R_1

SHR R_1



$$R_1 = 01100$$

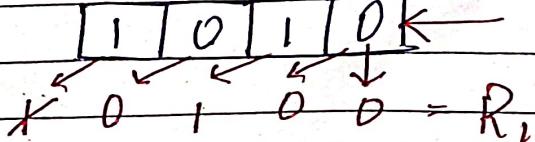


$$R_1 = 01010$$

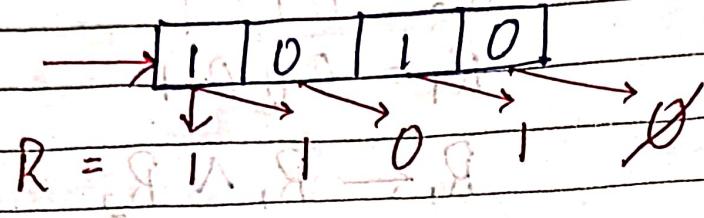
In Arithmetic shift-left, LGB is const.
Then the whole word will be shifted
left

$$R_1 = 1010$$

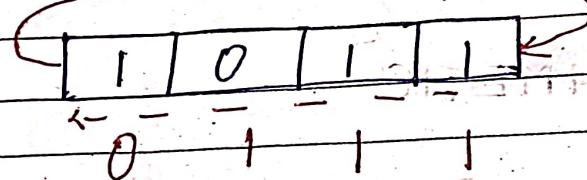
ASHL R_1



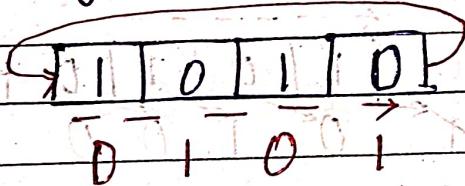
In Arithmetic Shift Right, MSB is const. Then The whole word will be shifted right.



In Circular Left, MSB comes to the LSB pos. Then There is shifting from right to left.



In circular right, LSB comes to the MSB pos. Then There is shifting from left to right.



* Add & Shift Algorithm (Booth Multiplication Algo)

=> Applicable for The no. where only one operand is even.

~~Rule~~LSB of BR \oplus

0 0 → aslrc

0 1 → Subtract, aslrc

1 0 → Add, aslrc

1 1 → aslrc

~~Q = -7 X 8~~

$$\begin{array}{r} 7 = 111 \\ 8 = 0100 \end{array}$$

$$7 = 111$$

$$8 = 1000$$

1st → Check The operands & choose The highest no. of bits

2nd → If n no. of highest bits Then add one more bit i.e. $n+1$

3rd → Make both The bits to be equal

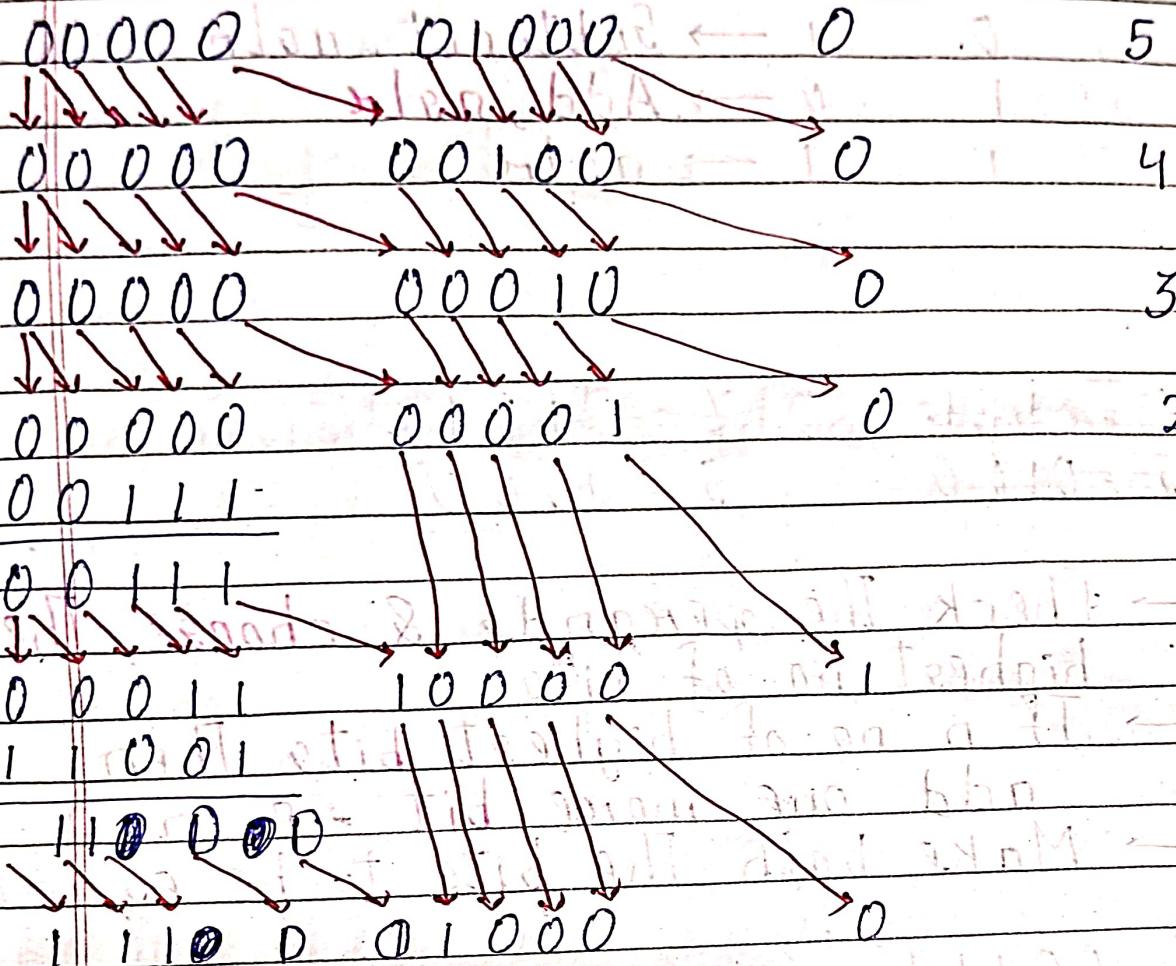
$$\begin{array}{r} -7 = 00111 \xrightarrow{1's} 11000 \xrightarrow{2's} 11001 \\ 8 = 01000 \end{array}$$

Initially, The content of AC is zero and fore n no. of bits Then n no. of zeroes.

Initially, The content of Q is zero.

If one is +ve & another -ve, Then The +ve no. should be taken by BR.

No. of bits in AC = no. of counts in SC

ACBRQSC

0001110001000

0000111001000 + 1

0000111000

32168

56

-7X-8

Q-

- + X 15
- 7 X - 13

Page No. / /

Date: / /

$$\begin{array}{l} -7 = 00111 \xrightarrow{1^g} 11000 \xrightarrow{2^g} 11001 \\ -8 = 01000 \xrightarrow{1^g} 10111 \xrightarrow{(-8)} 11000 \end{array}$$

AC BR Q GC

00000	11000	0	5
00000	01100	0	4
00000	00110	0	3
00000	00011	0	2
+00111			
00111			
00011	10001	1	1
10001	11000	1	0

(56)

AC BR Q GC

00000	11001	0	5
00000	01100	1	4
00000	11001	0	5
+01000			
01000			
00100	01100	1	4
11000	0		
11000			
11110	00110	0	3
11111	00001	0	2
01000			
00111			
00011	100001	1	1

Addressing Modes

AM	FA	content in AC	Add	Memory
Direct	500	800	PC = 200	200
Intermediate operand	201	500	R ₁ = 400	201
Indirect	800	300	X _R = 100	1202
			AC	
Relative	600	325		450
Indexed	600	900		700
Registers	400	500		800
Indexed	400	700	600	900
auto-decrement	399	450	702	325
auto-increment	400	700	800	300
ADD R ₁ , R ₂ ;	R ₁	$\leftarrow R_1 + R_2$		
ADD R ₁ , A ;	R ₁	$\leftarrow R_1 + M[A]$		

- Q- An instruction is stored at location 300 with its address field at location 301. The address field has the value 400. A processor register R₁ contains the no. 200. Evaluate the effective address if the addressing mode of the instruction is (a) direct (b) immediate (c) relative (d) register indirect (e) index with R₁ as the index register.

A-	<u>AM</u>	<u>EA</u>	Add.	Memory
Direct	400		300	Load to A/C
Immediate	301		301	Address 400
Relative	400 + 302 = 702	$R_1 = 200$	302	Next inst
Reg. indirect	200			
Index	400 + 200 = 600			

Q- A 2-word instruction each stored in memory at an address designated by the symbol 'w'. The address field of the instruction (stored at $w+1$) is designated by the symbol 'y'. The operand used during the execution of the instruction is stored at an address symbolized by 'z'. An index register contains the value x. State how z is calculated from the others.

Ques. Explain the architecture of cache system and its working mechanism. What are the various types of cache? Explain the working mechanism of cache system. What is the difference between L1 and L2 cache? Explain the working mechanism of cache system. What is the difference between L1 and L2 cache?

25/11/24

Page No. / /

Date: / /

Basic Instruction Format

15	14	12'11	0
I	opcode	ADDRESS	opcode = 000 through 110

a) Memory Reference Instruction

15	14	12'11	0
0	111	REGISTER OPERATION	opcode = 111 $J = 0$

b) Register Reference Instruction

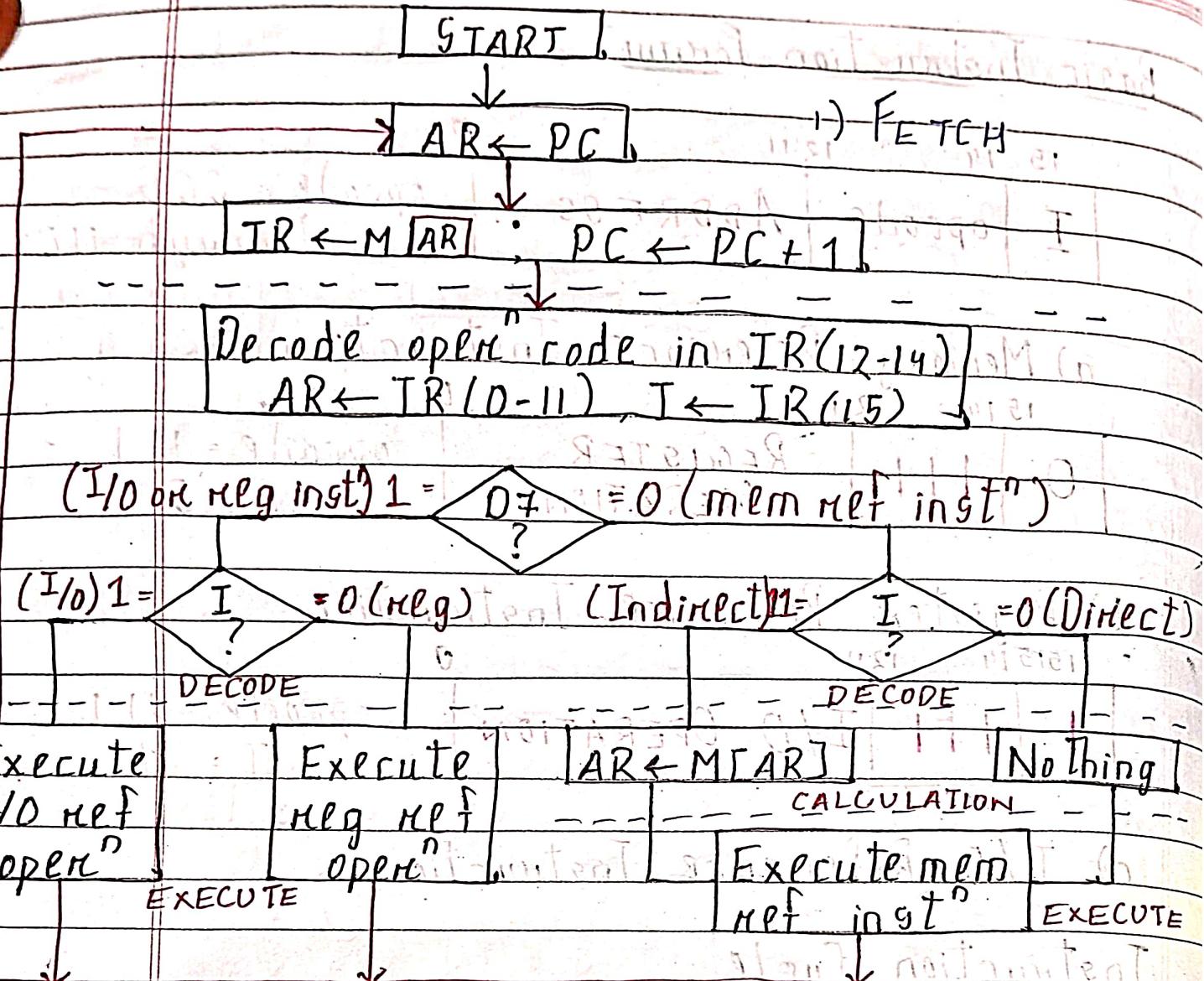
15	14	12'11	0
1	111	I/O OPERATION	opcode = 111 $J = 1$

c) I/O Reference Instruction

Instruction Cycle

- 1) Fetch
 - 2) Decode
 - 3) Execute
- Data resides in registers & I/O

- 1) Fetch
 - 2) Decode
 - 3) Calculation EA
 - 4) Execute
- Data resides in memory

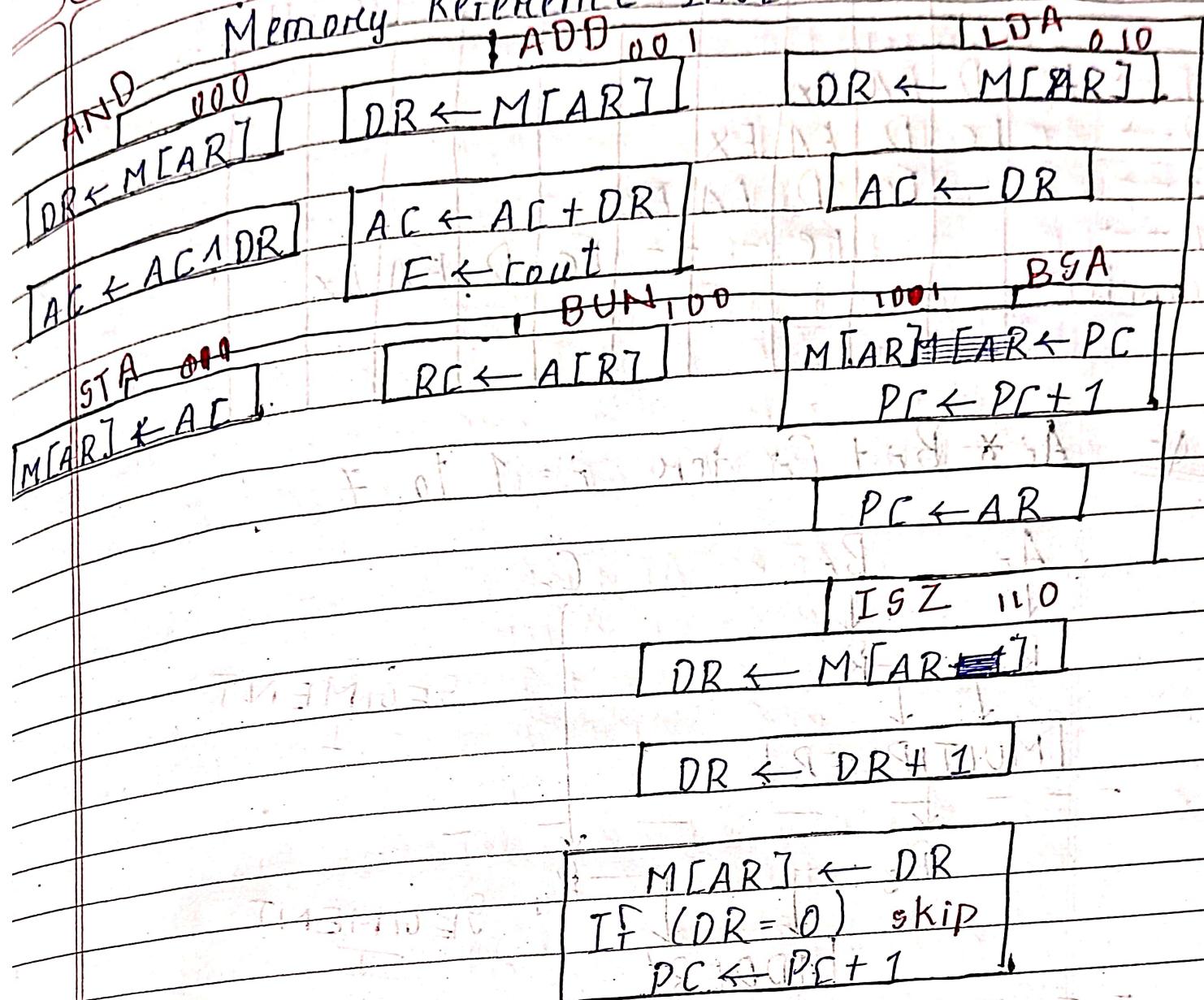


IGZ → Increment & skip if zero

Page No.

Date: / /

Memory Reference Instn



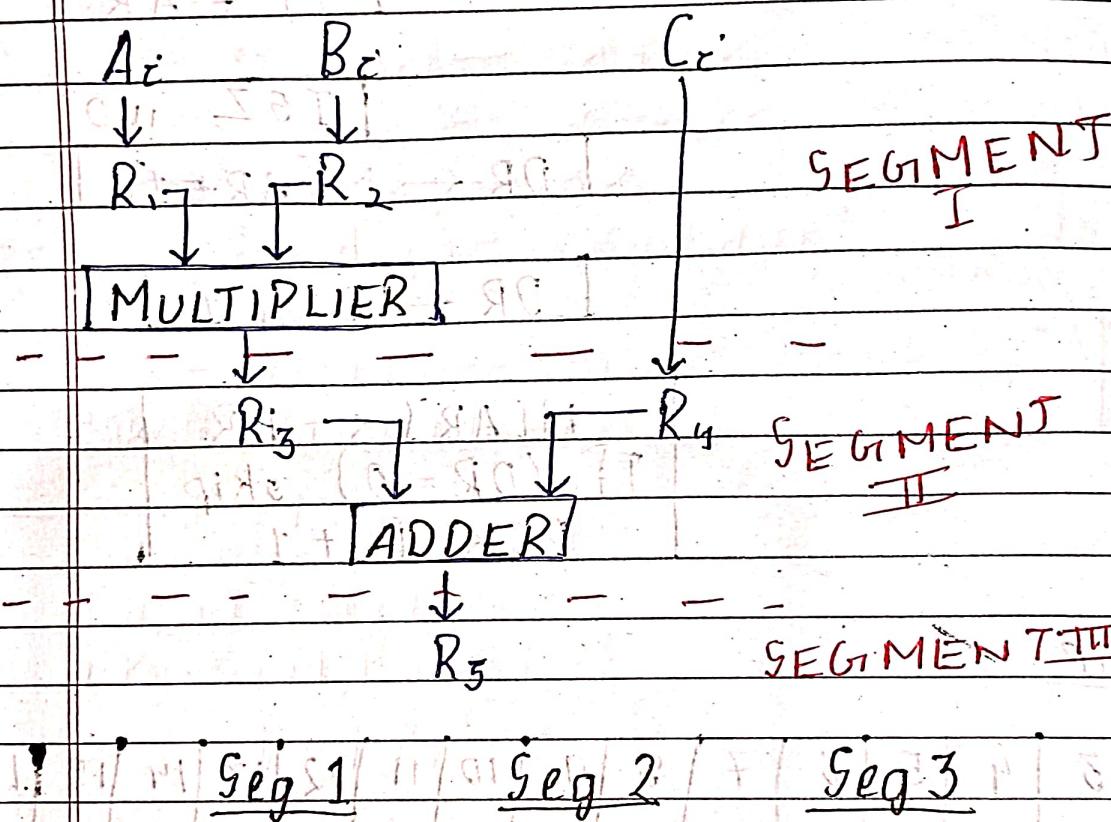
Pipeline

$C = n + k$ of segm
 $n = \text{no. of segm}$
 $k = \text{no. of int}$

Using Pipeline

	1	2	3	4	5	6	7	8	9	10	11	...
1 →	Fe	D	EA	Ex								
2 →		Fe	D	EA	Ex							
3 →			Fe	D	EA	Ex						
4 →				Fe	-	-	Fe	D	EA	Ex		
(branch)												

Q- $A_i * B_i + C_i$ for $i = 1$ to 7



	Seg 1	Seg 2	Seg 3
1	A_1, B_2	R_3, R_4	R_5
2	A_1, B_1		
3	A_2, B_2	$A_1 * B_1 + C_1$	-
4	A_3, B_3	$A_2 * B_2 + C_2$	$A_1 * B_1 + C_1$
5	A_4, B_4	$A_3 * B_3 + C_3$	$A_2 * B_2 + C_2$
6	A_5, B_5	$A_4 * B_4 + C_4$	$A_3 * B_3 + C_3$

Q-1

The 8 bit registers AR, BR, CR & DR initially have the following values

$$AR = 11110010$$

$$BR = 11111111$$

$$CR = 10111001$$

$$DR = 11101010$$

Determine the 8 bit values in each register after the execution of the following sequence of micro operations.

$$AR \leftarrow AR + BR$$

$$CR \leftarrow CR \wedge DR$$

$$BR \leftarrow BR + 1$$

$$AR \leftarrow AR - CR$$

A- $AR \rightarrow 11110010$ (ADD)

$$BR \rightarrow 11111111$$

$$\begin{array}{r} \times 11110001 \\ \hline 11110001 = AR \end{array}$$

$$CR \rightarrow 10111001$$

$$DR \rightarrow 11101010$$

$$\begin{array}{r} \hline 10101000 = CR \end{array}$$

$$BR \rightarrow 11111111$$

$$\begin{array}{r} + \\ \hline \times 00000000 = BR \end{array}$$

$$AR \rightarrow 11110010$$

$$CR \rightarrow 10101000$$

$$\begin{array}{r} \hline 01001001 = AR \end{array}$$

Q-2

An 8 bit register contains the binary value 10011100. What is the register value after an arithmetic shift right?

Starting from the initial no. 10011100, determine the register value after an arithmetic shift left and state whether there is an overflow.

A: $\rightarrow 10011100 \text{ ashrc}$
 $\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow$
 $11101110, 0 = R$

$10011100 \leftarrow \text{ashl}$
 $\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow$
 ~~$00011100 = R$~~

Yes, There is an overflow because -

- i) As There is 8 no. of bits but after shifting operation There is 9 bits.
- ii) The +ve no. changes to -ve no.

Q-3

Draw a space time diagram for a 6-segment pipeline showing the time it takes to process 8 tasks.

E/I K

Input

Output

↓

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆							
2	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆							
3	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆							
4	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆							
5	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆							
6	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆							
7	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆							
8	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆							

$$C = 6 + 8 - 1$$

$$\text{Ans} = 13$$

Q4

Starting from an initial value of R = 11011101, determine the sequence of binary values in R after a logical shift left, followed by a circular shift right, followed by a logical shift right, & a circular shift left.

A-