DIGITALS

NOTES

GATE 2009

JANARDHANARAO

Digitals

SUN. 17/08/08

Number Systems:

	Base/Radix	Numbers.
1. Decimal	10	0, 1, ... 9
2. Binary	2	0, 1
3. octal	8	0, 1, ... 7
4. Hexadecimal	16	0, 1, ... 9, A, B, C, D, E, F.

Each Hexa digit $\rightarrow$ 4 bits,

$3f_{16} \rightarrow 0011\ 1111_2$

Each octal digit $\rightarrow$ 3 bits

$316_8 \rightarrow 011\ 001\ 110_2$

Q. $110010_2 = x_{16}$

$\underset{3}{0011}\ \underset{2}{0010} = 32_{16}$

Q. $11011.01_2 = x_{16}$

$\underset{1}{0001}\ \underset{B}{1011}\ .\ \underset{4}{0100}$ $= 1B.4_{16}$

Q. $6728_{10} = x_2$

$6728_{10} \rightarrow 6728_{16} \rightarrow x_2$

$\Rightarrow$

16	6728	
16	420	– 8
16	26	– 4
	1	– 10 (A)

$1A48_{16}$

$= 0001\ 1010\ 0100\ 1000_2$

Q. Determine the possible bases of the following relations.

(1). $\sqrt{41} = 5$

max. digit is 5

so ~~max~~ min value of base is 6. So base ≥ 6

Let base = b.

$\sqrt{4 \times b^1 + 1 \times b^0}_{10} = 5 \times b^0_{10}$

$\Rightarrow \sqrt{4b+1} = 5$

$\Rightarrow 4b+1 = 25$

$\Rightarrow b = 6.$

Q. $\frac{302}{20} = 12.1$

Let base = b. Base ≥ 4 b'coz max digit is 3.

$\Rightarrow \frac{3b^2+2}{2b} = b+2+\frac{1}{b}$

$\Rightarrow \frac{3b^2+2}{2\cancel{b}} = \frac{b^2+2b+1}{\cancel{b}}$

$\Rightarrow b = 4.$

Q. $\frac{44}{4} = 11$

Let base = b. Observed base ≥ 5, b'coz maximum value of digit = 4.

$\frac{4b+4}{4} = b+1 \Rightarrow b+1 = b+1$

The above relation is valid in all the no. system with base ≥ 5.

Q. In a positional weight system x & y are two successive digits and $xy = 25_{10}$ & $yx = 31_{10}$. Determine the values of base x & y.

Here $b = ?$, $x = ?$ & $y = ?$

and $y = x+1$.

$(x)(x+1) = 25_{10}$ $\quad ((x+1)b+x)_{10} = 31_{10}$

$\Rightarrow [x \times b + (x+1) = 25]_{10}$ $\quad \Rightarrow x(b+1)+b = 31 \rightarrow (2)$

$\Rightarrow x(b+1)+1 = 25 \rightarrow (1)$

(1) - (2) $\Rightarrow b = 7$. Then from (1) $\Rightarrow x = 3, y = 4$.

Complementary Number Representation :-

Base = r

⇒ (r−1)'s complement

⇒ r's complement

Decimal System (r = 10).

9's complement of 168_{10} ⇒

$$\begin{array}{r} 999 \\ (-)\ 168 \\ \hline 831_{10} \end{array}$$

10's complement of 168_{10} ⇒ 9's comp + 1

$$\Rightarrow \begin{array}{r} 999 \\ (-)\ 168 \\ \hline 831 \end{array} + 1 = 832_{10}$$

Q.
$$\begin{array}{r} 862_{10} \\ (-)\ 491_{10} \\ \hline ? \end{array} = 862_{10} + (-491_{10})$$

(i). 862 + (9's of 491) ⇒

$$\begin{array}{r} 862 \\ (+)\ 508 \\ \hline 1\ 370 \\ +1 \\ \hline 371 \end{array}$$

+1 → EOC

(ii). 862 + (10's of 491) =

$$\begin{array}{r} 862 \\ 509 \\ \hline 1\ 371_{10} \end{array}$$

EOC 1 ↑ ignore

Q.
$$\begin{array}{r} 491_{10} \\ -\ 862_{10} \\ \hline ? \\ -371_{10} \end{array} = 491 + (-862) = \begin{array}{r} 491 \\ (+)\ 138 \\ \hline 629 \end{array} \leftarrow 10\text{'s}$$

NO EOC ↓ − (−ve sign)

629 ↓ 10's → 371

Digital System (r = 2)

1's complement of 1011 → 0100_2

2's complement of 1011 → 1's of 1011 + 1

→ 0100 + 1 = 0101

Q. $x = 1000111000$

2's complement of x = 0111001000

Q. $x = 1011$

2's of x = 0101

Q. $11010_2 - 01110_2$ = 11010 + (− 01110)

(i). 11010 + (1's of 01110) = 11010 + 10001

$$\begin{array}{r} 11010 \\ +\ 10001 \\ \hline \text{EOC } 1\ 01011 \\ +1 \\ \hline 01100 \end{array}$$

(ii). 11010 + (2's of 01110)

$$= \begin{array}{r} 11010 \\ +\ 10010 \\ \hline 1\ 01100 \end{array}$$

EOC ignore

Q. $01110_2 - 11010_2$ = 01110 + (2's of 11010)

$$= \begin{array}{r} 01110 \\ +\ 00110 \\ \hline 10100 \end{array}$$

NO EOC → Negative sign

10100 ↓ 2's

01100

$2^4 = 16$

$16 - 2 = 14$

$16 - 1 = 15$

1's comp	2's comp
+0 = 0000	+0 = 0000
−0 = 1's comp of +0 = 1's of 0000 = 1111	−0 = 2's comp. of +0 = 0000

(Disadv. of 1's complement)

* Range of numbers represented using 'n' bits

Go represent (16) numbers

1's comp. form $\rightarrow +(2^{n-1}-1)$ to $-(2^{n-1}-1)$

let $n=4 \Rightarrow +7$ to $-7 \longrightarrow (14)$

2's comp. form $\Rightarrow +(2^{n-1}-1)$ to -2^{n-1}

let $n=4 \Rightarrow +7$ to $-8 \longrightarrow (15)$

Q. How many bits are required to represent -64_{10} in a). 1's comp. form b). 2's form

1's form $\Rightarrow +(2^{n-1}-1)$ to $-(2^{n-1}-1)$

let $n=7 \Rightarrow +63$ to -63

✓ $n=8 \Rightarrow +127$ to -127.

2's form $\Rightarrow +(2^{n-1}-1)$ to -2^{n-1}

✓ let $n=7 \Rightarrow +63$ to -64

Q. 10's comp for $(731)_{11}$

$$\begin{array}{rccc} & A & A & A \\ & 7 & 3 & 1 \\ (-) & & & \\ \hline & 3 & 7 & 9 \end{array}$$

Q. 9's comp of $(731)_{10}$

$$\begin{array}{rc} & 999 \\ (-) & 731 \\ \hline & 268 \end{array}$$

Binary Numbers :

(a). Unsigned Numbers $\longrightarrow$ [n bits] — magnitude

(b). Signed Numbers $\longrightarrow$ [MSB |] — sign bit, magnitude

↓ represented by

(i). sign magnitude

(ii). 1's comp form

(iii). 2's comp form

Sign bit: $0 \rightarrow +ve$, $1 \rightarrow -ve$

These three representations are same for unsigned (+ve Numbers).

(i). sign magnitude $\rightarrow$ $+3 = 0\underline{11}$
$-3 = 1\underline{11}$

(ii). 1's comp. form $\rightarrow$ $+3 = 011$
$-3 = \underline{\text{1's comp of } +3}$
$= 100$

(iii). 2's comp. form $\rightarrow$ $+3 = 011$
$-3 = $ 2's comp of $+3$
$= 101$

Q. Decimal equivalent of 2's number $\underline{1}01$ is $-$?

2's

$\underline{-011}$
$= -3_{10}$

Q. Decimal equivalent of sign mag. no. 111 is $-$?
-3_{10}

Q. Represent $+53_{10}$ & -53_{10} in all the 3 forms of signed no. representation.

$$53_{10} \rightarrow \begin{array}{r|l} 2 & 53 \\ 2 & 26 - 1 \\ 2 & 13 - 0 \\ 2 & 6 - 1 \\ 2 & 3 - 0 \\ & 1 - 1 \end{array} = 110101_2$$

$+53 = 0110101$

	Sign mag. form	1's form	2's form
$+53_{10}$	$0\,\underline{110101}$	0110101	0110101
-53_{10}	$1\,\underline{110101}$	$-53 =$ 1's of $+53$ $= 1001010$	$-53 =$ 2's of $+53$ 1001011

Q. What are the decimal equivalents of the following signed no.s in all the 3 forms.

	Sign mag. form	1's form	2's form
01101	$+13_{10}$	$+13_{10}$	$+13_{10}$
101010	$\underline{1}\ \overline{01010}$ → -10_{10}	1 01010 ↓1's → -010101 $= -21_{10}$	1 01010 ↓2's → -010110 $= -22_{10}$
111111	$\underline{1}\ \overline{11111}$ $= -31_{10}$	1 11111 ↓ ↓1's → -0	1 11111 ↓ ↓2's → -1

Q. Decimal equivalent of 2's no. 1000 is — ?

1 000

↓ ↓2's

-1000

$= -8_{10}$

Q. Decimal equivalent of 2's no. 10000 is — ?

1 0000

↓2's

-10000

$= -16_{10}$

Q. What is the equivalent 2's comp representation of a 2's comp. no. 1101 is — ?

(a). 00<u>1101</u> (b). 0<u>1101</u> (c). 10<u>1101</u> ✓(d). 11<u>1101</u>

$+6 = 0110$

$-6 = 2\text{'s of } +6 = 2\text{'s of } 0110$

$= 1010$

$= 2\text{'s of } 00110 = 11010$

$= 2\text{'s of } 000110 = 111010$

Q. A Register contains a 2's comp. no. 10110. What is the content of the register if it is divided by 2.

decimal equi. of 10110 = − 01010

$= -10_{10}$; $\frac{-10}{2} = -5$

−5 = 2's of +5

= 2's of 00101 = 11011

(or).

$\boxed{10110}$ (−10) $\xrightarrow{\div 2}$ $\boxed{11011}$ (−5) $\xrightarrow{\div 2}$ $\boxed{11101.1}$ (−2.5)

$\downarrow$ 2's

00010.1 $= -2.5_{10}$

Binary codes :-

a). Alpha numeric
- ASCII code [7 bits, $2^7 = 128$ Alpha numericals]
- EBCDIC [8 bits, $2^8 = 256$ Alpha numericals]

b). Numeric — BCD (0–9) → each decimal digit → 4 bits

- weighted
  - Natural BCD code
  - positively ↓ Eg: 8 4 2 1, 2 4 2 1, 3 3 2 1
  - Negatively ↓ 6 3 1 −1 → sum of weights = 9.
- Non-weighted
  - Eg: Gray code, Excess−3 code

(2 4 2 1, 3 3 2 1, 6 3 1 −1) → Self complementary codes

Excess−3 : self complementary code, sequential code.

8421 : sequential code.

Dec. digit	Natural BCD 8421	Excess-3	2421 (self comple)	631−1 (self comple)	Gray
0	0000	0011	0000	0000	0000
1	0001	0100	0001	0010	0001
2	0010	0101	0010	0101	0011
3	0011	0110	0011	0100	0010
4	0100	0111	0100	0110	0110
5	0101	1000	1011	1001	0111
6	0110	1001	1100	1011	0101
7	0111	1010	1101	1010	0100
8	1000	1011	1110	1101	1100
9	1001	1100	1111	1111	1101

743_{10} in (1). BCD → $0111\ 0100\ 0011_{BCD}$.

(2). 3321 → $1101\ 0101\ 0011_{3321}$

3 ↓ 1000, 0100, 0011 (3321) } Self complementary

7 ↳ 2 = 0010 (3321)

Now 7 ⇒ comp. of 0010 = 1101.

(3). Binary, → $2^n \geq 743$, $n = 10$.

16 | 743
16 | 46 − 7
16 | 2 − 14 (E)

$2E7_{16} = 0010\ 1110\ 0111_2$

Gray code: (Reflective code, unit distance code)

1-bit	2-bit	3-bit
0	00	000
1	01	001
	11	011
	10	010
		110
		111
		101
		100

→ differ by 1-bit

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$
Modulo-2 Addition (Exclusive OR)

Binary : 1 0 1 1 0
↓ ⊕ ⊕ ⊕ ⊕
Gray : 1 1 1 0 1
↓ ⊕ ⊕ ⊕ ⊕
Binary : 1 0 1 1 0

BCD Addition :-

$$\begin{array}{rl} 6_{10} & 0110_{BCD} \\ +\ 2_{10} = & 0010_{BCD} \\ \hline & 1000_{BCD} \\ & \downarrow 8_{10} \end{array}$$

$$\begin{array}{rll} 8_{10} & & 1000_{BCD} \\ +\ 6_{10} = & & 0110_{BCD} \\ \hline & & 1110 \rightarrow \text{not a valid BCD.} \\ & & +\ 0110 \\ \hline & 0001 & 0100_{BCD} \\ & 14_{10} & \end{array}$$

$$\begin{array}{rll} 9_{10} & & 1001_{BCD} \\ +\ 8_{10} = & & 1000_{BCD} \\ \hline & 1 & 0001 \\ & & +\ \boxed{0110} \\ \hline & 1 & 0111_{BCD} \\ & 17_{10} & \end{array}$$

Decimal	Binary/Hexa
0	0
1	1
2	2
3	3
4	4
5	:
6	8
:	9
10	A
11	B
12	C
	D
	E
	F
	10
	11

8 + 2 = 10

10 ← A

(Diff = 6)

16 ← 10

6

Q. In the following BCD additions how many BCD corrections are required.

$$\begin{array}{rl} 49_{10} & 0100\ 1001 \\ +\ 57_{10} = & 0101\ 0111 \\ \hline & 1010\ 0000 \\ & +\ 0110\ 0110 \\ \hline & 1\ \ 0000\ 0110 \\ & 1\quad 0\quad 6_{10} \end{array}$$

Ans: 2 times.

Q. $176_{10} + 824_{10}$

$$\begin{array}{r} 0001\ 0111\ 0110 \\ 1000\ 0010\ 0100 \\ \hline 1001\ 1001\ 1010 \\ 0110 \\ \hline 1001\ 1010\ 0000 \\ 0110\ \quad \\ \hline 1010\ 0000\ 0000 \\ 0110\ \quad\quad \\ \hline \ldots 0000\ 0000\ 0000 \end{array}$$

Ans: 3 times

$$\begin{array}{r} 176 \\ 824 \\ \hline 1000 \end{array}$$

* SUNDAY. 31. Aug. 2008 *

Boolean Algebra :

AND Law	OR Law
$A \cdot 0 = 0$	$A + 0 = A$ ← Identity element.
$A \cdot 1 = A$	$A + 1 = 1$
Identity element $A \cdot A = A$	$A + A = A$
$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$

(1). Commutative law:

$A + B = B + A$

$A \cdot B = B \cdot A$

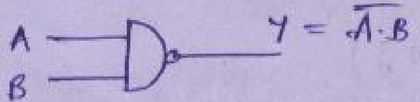
* AND, OR operations are commutative & Associative

(2). Associative law:

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

$(A \cdot B) \cdot C = A \cdot (B \cdot C)$

Q find the commutative & Associative operations of NAND.



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

(b). $(\overline{A \cdot B})$ NAND $C = \overline{\overline{A \cdot B} \cdot C}$

A NAND (B NAND C) $=$ A NAND$(\overline{B \cdot C})$

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

$\longrightarrow \overline{\overline{A \cdot B} \cdot C} \neq \overline{A \cdot \overline{B \cdot C}}$

* NAND operation is commutative but not associative.

(3). Distribution law :

$A \cdot (B+C) = AB + AC$

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

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

$= (A+B)$.

(ii). $\bar{A} + AB = (\bar{A} + A)(\bar{A} + B)$

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

(4). Consensus Law:

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

Eg: $yx + \bar{y}z + pwxz = yx + \bar{y}z$

proof: $AB + \bar{A}C + BC(A + \bar{A})$

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

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

$= AB + \bar{A}C.$

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

(5). Transposition law:

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

Eg: $xy + \bar{y}z = (x+\bar{y})(y+z)$

RHS: $(x+\bar{y})(y+z) = xy + z\bar{y} + xz$

$= xy + \bar{y}z.$

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

(6). De Morgan's law:

$\overline{A + B + C + \cdots} = \bar{A}\cdot\bar{B}\cdot\bar{C}\cdot\cdots$

$\overline{A\cdot B\cdot C\cdots\cdots} = \bar{A} + \bar{B} + \bar{C} + \cdots$

Additional laws:

(1). $x\cdot f(x, \bar{x}, w, y, \ldots z)$

$= x\cdot f(1, 0, w, y, \cdots z)$

$$x + f(x, \bar{x}, \omega, y, \ldots z)$$
$$= x + f(0, 1, \omega, y, \ldots z)$$

(7). Duality:

All the boolean expressions resulting from interchanging of operators and identity elements are valid.

Eg: $A \cdot 1 = A$
$\Rightarrow A + 0 = A$

Adv: To findout complement of a function f.

Step 1: find Dual of f ie f_D.

Step 2: Compliment of all var.s $\rightarrow \bar{f}$.

Eg: $A + B + CD$

$f_D = A \cdot B \cdot (C + D)$

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

$= \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot \bar{D}$.

$\bar{f} = \overline{A + B + CD}$

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

$= \bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot \bar{D}$.

Q. Simplify following Boolean functions.

(1). $f = AB + \bar{A}C + \bar{C}D + \bar{B}C$

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

$= \underbrace{AB}_{x} + \underbrace{\overline{AB}}_{\bar{x}}C + \bar{C}D$

$= AB + (C + \bar{C}D)$

$= AB + C + D$.

(2). $f = AB\bar{C} + A\bar{B}C + \bar{A}BC + ABC$

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

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

$= AB + BC + AC$.

(3). $f = \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz + xyz$

$= \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz + \bar{x}yz + \bar{x}yz + xyz$

$= \bar{x}z(\bar{y}+y) + \bar{x}y(\bar{z}+z) + yz(\bar{x}+x)$

$= \bar{x}z + \bar{x}y + yz$.

Q. How many two i/p NAND's are required to implement the following

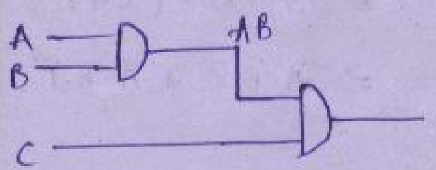
(i). $f(A,B,C) = A + AB + ABC$.

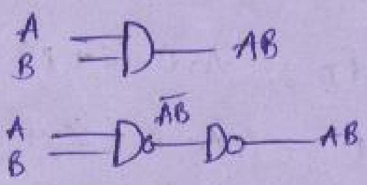
$= A + AB(1+C)$

$= A + AB = A$.

Ans: zero NAND Gates.

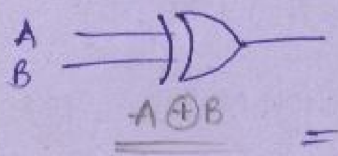
(ii). $f = ABC$.





Each AND is replaced by two NAND's. So the total no. of NAND Gates = 4.

Q. Implement Ex-OR using min. no. of NAND gates.



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

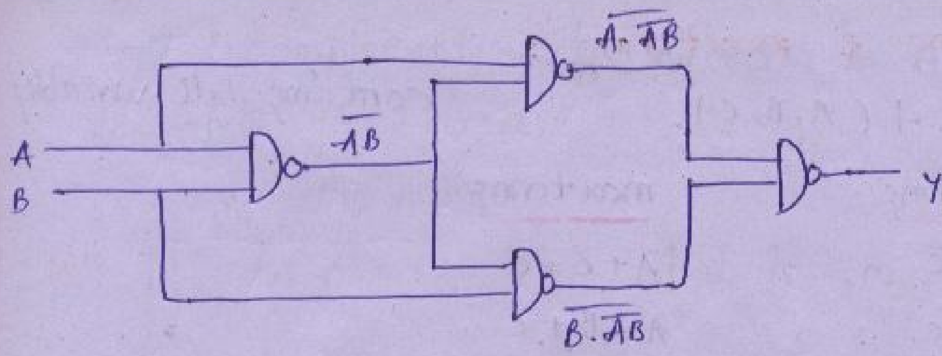
$= \bar{A}B + A\bar{B} + A\bar{A} + B\bar{B}$

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

$= A\,\overline{AB} + B\cdot\overline{AB}$

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

$= \overline{(\overline{A\cdot\overline{AB}})\cdot(\overline{B\cdot\overline{AB}})} \Rightarrow$ 5 NAND's.



Here NAND's are replaced by NOR's then we get Ex- NOR gate:

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

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

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

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

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

Operator precedence :

(1). paranthesis ( )

(2). NOT

(3). AND $\cdot$

(4). OR $+$

Literal = variable (or) complement of a var.

Implement x - NOR using min. no. of NOR's.

minterms & maxterms :

Let $f(A,B,C)$. contains all variables

minterms	maxterms
$\bar{A}\cdot\bar{B}\cdot\bar{C}$	$\bar{A}+\bar{B}+\bar{C}$
$\bar{A}\cdot\bar{B}\cdot C$	$\bar{A}+\bar{B}+C$
8 $\bar{A}\cdot B\cdot\bar{C}$	8 $\bar{A}+B+\bar{C}$
⋮	⋮
$AB\cdot\bar{C}$	$A+B+\bar{C}$
ABC	$A+B+C$

* for 'n' var. function → 2^n minterms
2^n maxterms

* Sum of all minterms = 1. $\sum_{i=0}^{2^n-1} m_i = 1$

* product of all maxterms = 0. $\prod_{i=0}^{2^n-1} M_i = 0$

* product of any two minterms = 0.

$$m_i \cdot m_j = 0, \text{ if } i \neq j$$
$$= m_i, \text{ if } i = j$$

* Sum of any two maxterms = 1.

$$M_i + M_j = 1, \text{ if } i \neq j$$
$$= M_i, \text{ if } i = j.$$

Let $f(x,y)$

minterm: $1 = \text{var}$, $0 = \overline{\text{var}}$ — maxterm: $1 = \overline{\text{var}}$, $0 = \text{var}$

x	y	minterm		max term	
0	0	$\bar{x}\cdot\bar{y}$	m_0	$x+y$	M_0
0	1	$\bar{x}\cdot y$	m_1	$x+\bar{y}$	M_1
1	0	$x\cdot\bar{y}$	m_2	$\bar{x}+y$	M_2
1	1	xy	m_3	$\bar{x}+\bar{y}$	M_3

⇒ complement of minterm = maxterm and vice-versa.

$M_j = \overline{m_j}$

Q. If $f(A,B,C,D,E)$. what is $m_{23} = ?$

$m_{19} = ?$ $M_{28} = ?$, $M_{23} = ?$

$23 \rightarrow 10111$ $19 \rightarrow 10011$

$m_{23} = A.\bar{B}.C.D.E$ $m_{19} \rightarrow A.\bar{B}.\bar{C}.D.E$

$28 \rightarrow 11100$ $23 \rightarrow 10111$

$M_{28} \rightarrow \bar{A}+\bar{B}+\bar{C}+D+E$ $M_{23} = \bar{A}+B+\bar{C}+\bar{D}+\bar{E}$

$M_{23} = \overline{m_{23}} = \overline{A.\bar{B}.C.D.E}$

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

Q. $A \oplus A \oplus A \cdots \oplus A = ?$

$A \oplus A \oplus A \oplus A$, if even no. of A's.

$= 0 \oplus 0 = 0$

$A \oplus A \oplus A$, if odd no. of A's.

$= 0 \oplus A = A$

* ∴ $A \oplus A \oplus A \cdots \oplus A = 0$, if no. of terms = even

$= A$, if " = odd

* $\bar{A} \oplus \bar{A} \oplus \bar{A} \oplus \cdots \oplus \bar{A} = 0$, if no. of terms = Even

$= \bar{A}$, " = odd

Q. How many Boolean fun's are possible, using 'n'-var's

using n-var's $\longrightarrow \underset{(x)}{2^n}$ minterms

x minterms can be arranged in 2^x ways.

ie 2^{2^n} boolean functions are possible.

for 2 var.s $\rightarrow$ $2^{2^2} = 16$ functions.

$f(x, y)$.

	x	y	f_1	f_2	f_3	. . . . . .	f_{16}
m_0	0	0	0	0	0		1
m_1	0	1	0	0	0		1
m_2	1	0	0	0	1	. . . . . .	1
m_3	1	1	0	1	0		1
			ϕ	AND	(Inhibition) $x\bar{y} = x/y$	. . . .	1

Algebraic forms of Boolean functions:

①. Standard form $<$ stand SOP form / stand. POS form

②. Canonical form $<$ Cano. SOP form (or) Sum of minterms / Cano. POS form (or) product of maxterms

$f_1(A, B, C) = (A+B+\bar{C})(A+\bar{B}+\bar{C}) \rightarrow$ Cano. POS

$f_2(A, B, C) = AB\bar{C} + ABC + \bar{A}\bar{C} \rightarrow$ stand. SOP form

* SAT. 11/10/08 *

Q. Convert the following boolean eq. into canonical SOP form.

1). $f(A, B, C) = \bar{A} + A\bar{B}C + B\bar{C}$ — Std. SOP

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

$= \underset{011}{\bar{A}BC} + \underset{010}{\bar{A}B\bar{C}} + \underset{001}{\bar{A}\bar{B}C} + \underset{000}{\bar{A}\bar{B}\bar{C}} + \underset{101}{A\bar{B}C} + \underset{110}{AB\bar{C}}$

$+ \underset{010}{\bar{A}B\bar{C}}$

$= m_3 + m_2 + m_1 + m_0 + m_5 + m_6$

$= \Sigma m(0, 1, 2, 3, 5, 6)$

(OR)

$\bar{A}$

A	B	C
0	–	–

$\downarrow$

$000 \rightarrow m_0$
$001 \rightarrow m_1$
$010 \rightarrow m_2$
$011 \rightarrow m_3$

$B\bar{C}$

A	B	C
ϕ	1	0

$010 \rightarrow m_2$
$110 \rightarrow m_6$

$\overset{1\,0\,1}{A\bar{B}C} \rightarrow m_5$

$f = \Sigma m(0,1,2,3,5,6) \rightarrow$ cano. SOP

$f = \pi M(4,7) \longrightarrow$ cano. POS.

Q. Convert the following boolean eq. into cano. POS form.

$f(A,B,C) = \bar{A}\cdot(\bar{B}+\bar{C})\cdot\underbrace{(\overset{0}{A}+\overset{0}{B}+\overset{1}{\bar{C}})}_{M_1} \rightarrow$ std. POS form

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

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

$= (\overset{1}{\bar{A}}+\overset{0}{B}+\overset{0}{C})(\overset{1}{\bar{A}}+\overset{1}{\bar{B}}+\overset{0}{C})(\overset{1}{\bar{A}}+\overset{0}{B}+\overset{1}{\bar{C}})(\overset{1}{\bar{A}}+\overset{1}{\bar{B}}+\overset{1}{\bar{C}})$
$(\overset{0}{A}+\overset{1}{\bar{B}}+\overset{1}{\bar{C}})(\overset{1}{\bar{A}}+\overset{1}{\bar{B}}+\overset{1}{\bar{C}})(\overset{0}{A}+\overset{0}{B}+\overset{1}{\bar{C}})$

$= M_4\cdot M_6\cdot M_5\cdot M_7\cdot M_3\cdot M_7\cdot M_1$

$= \pi M(1,3,4,5,6,7) \longrightarrow$ cano. POS

$= \Sigma m(0,2) \longrightarrow$ cano. SOP.

[OR]

A	B	C
1	–	–

$\Downarrow$

$1\,0\,0 \rightarrow M_4$
$1\,0\,1 \rightarrow M_5$
$1\,1\,0 \rightarrow M_6$
$1\,1\,1 \rightarrow M_7$

A	B	C
–	1	1

$\Downarrow$

$0\,1\,1 \rightarrow M_3$
$1\,1\,1 \rightarrow M_7$

Q. convert the following into cano. pos form.

$f(x,y,z) = x y + \bar{x} z \longrightarrow$ std. SOP

std. SOP → std pos, cano. sop

std pos → cano. pos

cano. sop → cano. pos

$\Rightarrow f = (x+z)(\bar{x}+y)$

↑ std pos

x	y	z
0	–	0

x	y	z
1	0	–

M_0 0 0 0   1 0 0 M_4

M_2 0 1 0   1 0 1 M_5

$f = \pi M(0,2,4,5) \longrightarrow$ cano. pos

K- maps :-

2- variable k-map

A \ B	0	1
0	0	1
1	2	3

neighbours

$m_0 \longrightarrow m_1, m_2$

$m_2 \longrightarrow m_0, m_3$

3 - var. k- map

Gray code — to make neibours differ by only one bit

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

neighbours

$m_0 \longrightarrow m_1, m_4, m_2$

$m_6 \longrightarrow m_2, m_7, m_4$

4 var. k- map

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

neighbours :

$m_0 \longrightarrow m_1, m_4, m_2, m_8$

$m_9 \longrightarrow m_8, m_{11}, m_{13}, m_1$.

group of 8 → octet

group of 4 → quad

group of 2 → pair

↦ single minterm

In 3-var. k-map : Quads: 0145, 1357, 3276, 0246, 0132, 4576 : Total = 6.

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

[That is from cano SOP into std SOP].

A \ BC	00	01	11	10
0	1		1	1
1	1	1		1

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

In 4-var. k-map :

Total Octets = 8 ; columns 12, 23, 34, 41 ; Rows 12, 23, 34, 41

Q. Simplify $f(A,B,C,D) = \Sigma m(0,2,3,5,6,10,4,11,12,13,15)$.

AB \ CD	00	01	11	10
00	1		1	1
01	1	1		1
11	1	1	1	
10			1	1

$\Rightarrow f = B\bar{C} + \bar{A}\bar{D} + \bar{B}C + ACD$.

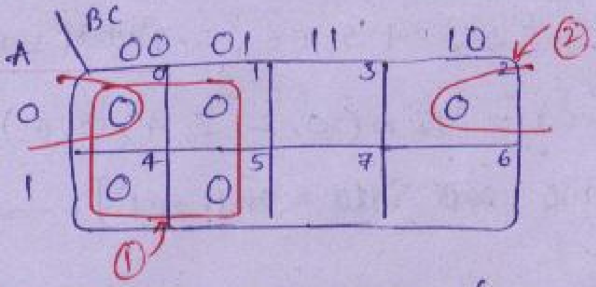
Simplified k-map eq. is a minimal eq but not unique.

Q. Simplify $f(w,x,y,z) = \Sigma m(0,1,2,4,5,8,9,10,12,14,15)$

wx \ yz	00	01	11	10
00	1	1		1
01	1	1		
11	1		1	1
10	1	1		1

$f = \bar{y}\bar{z} + \bar{w}\bar{y} + \bar{x}\bar{z} + \bar{x}\bar{y} + wxy$

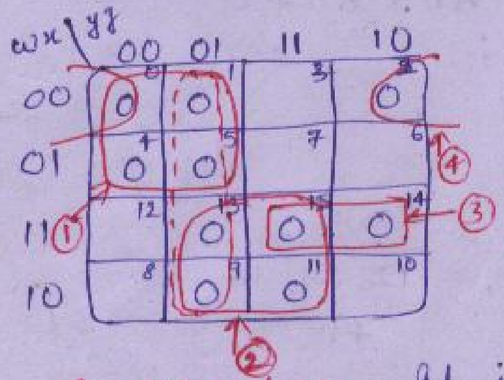
Q. $f(A,B,C) = \pi M(0,1,2,4,5,\cancel{6}) \rightarrow$ cano. pos



{ convert it into std pos form }

$f = B \cdot (A+C)$.

Q. $f(w,x,y,z) = \pi M(0,1,2,4,5,9,11,13,14,15)$.



$f = (w+y)(\bar{w}+\bar{z})(\bar{w}+\bar{x}+\bar{y})$
$(\cancel{x}+w+z)$

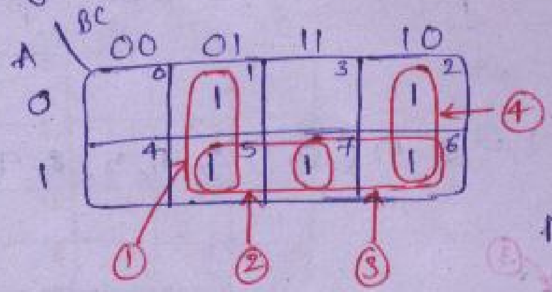
Implicant : It indicates the set of all adjacent minterms.

prime Implicant : It is an implicant which is not a subset of another implicant.

Essential PI : It is a PI which covers minterms exclusively.

Minimal expressions = EPI's + PI's (optional)

Eg: $f(A,B,C) = \Sigma m(1,2,5,6,7)$



All are PI's.

EPI's = ①, ④

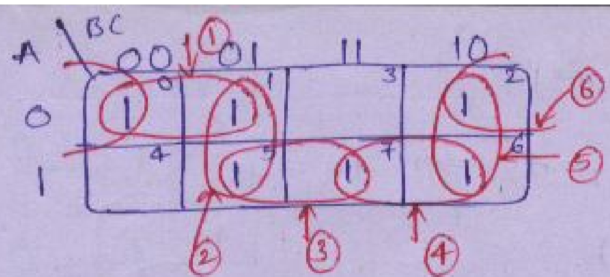
Minimal expressions

= ① + ④ + ②

(or) ① + ④ + ③

Q. Find EPI's and minimal expressions for the following boolean functions.

$f(A,B,C) = \Sigma m(0,1,2,5,6,7)$



EPI's = 0 (Nil)

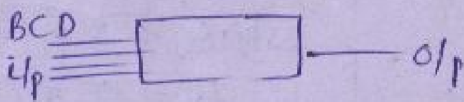
Minimal expressions = ① + ③ + ⑤

(or) ② + ④ + ⑥

Dont care conditions :-

For non-occuring i/p's the o/p can be assumed as 0 or 1. and this is called as Dont care condition.

Eg :



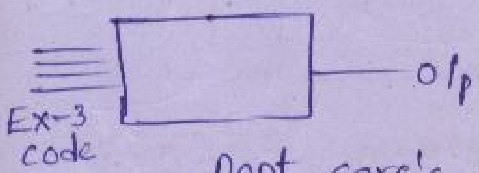
Valid BCD i/p's

0 – 0000
1 – 0001
:
9 – 1001

Non-occuring i/p's. o/p

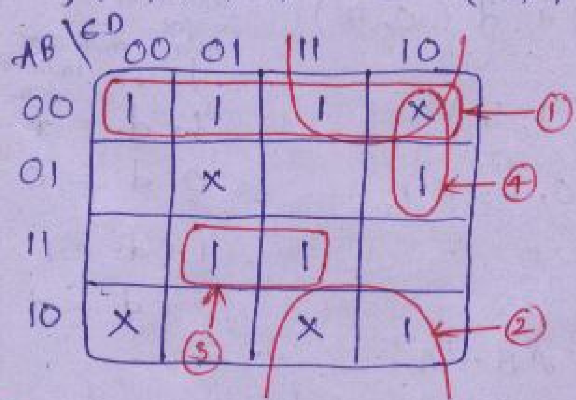
10 – 1010 ⟶ X
11 – 1011 ⟶ X
12 – 1100 ⟶ X
13 – 1101 ⟶ X
14 – 1110 ⟶ X
15 – 1111 ⟶ X

} dont care's



Dont care's : 0000 ⟶ X
0001 ⟶ X
0010 ⟶ X

Q. $f(A,B,C,D) = \Sigma m(0,1,3,6,10,13,15) + d(2,5,8,11)$



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

Q. $f_1 = \Sigma m(0,2,4,7)$; $f_2 = \Sigma m(1,2,4,6)$

$f = f_1 \cdot f_2 \Rightarrow f = ?$

$f = \Sigma m(2, 4)$

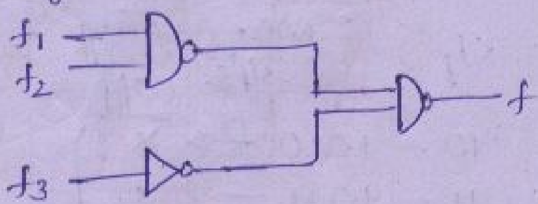
Similarly $f_3 = f_1 - f_2 \Rightarrow f_3 = ?$

$f_3 = \Sigma m(0, 7)$.

$f_4 = f_2 - f_1 \Rightarrow f_4 = ?$

$f_4 = \Sigma m(1, 6)$.

Q. Determine the function f_3 in the following logic ckt.



where $f = \Sigma m(0,1,3,5)$

$f_1 = \Sigma m(2,3,6,7)$

$f_2 = \Sigma m(0,1,5)$.

$f = \overline{\overline{f_1 f_2} \cdot \overline{f_3}}$

$= f_1 f_2 + f_3$

$\Rightarrow f_3 = f - f_1 \cdot f_2$

But $f_1 \cdot f_2 = \phi$

$\Rightarrow f_3 = f = \Sigma m(0,1,3,5)$.

Q. $f = f_1 \cdot f_2$ where $f_1 = \Sigma m(0,1,5) + d(2,3,7)$

$f_2 = \Sigma m(1,2,4,5) + d(0,7)$.

$f = f_1 \cdot f_2 = \Sigma m(1,5) + d(0,2,7)$.

minterm in one fun. — dont care in another fun.

$1 \cdot d = d$

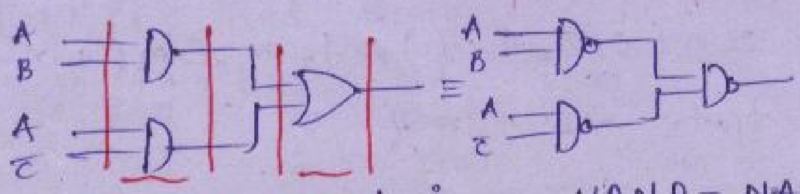
$0 \cdot d = 0$

$1 + d = 1$

$0 + d = d$

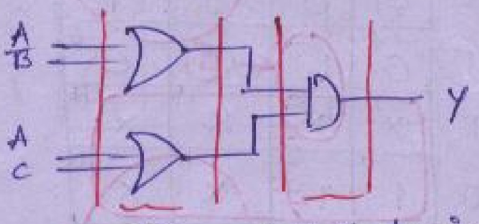
Two level logic :-

SOP form $\longrightarrow$ $Y = AB + A\bar{C}$



AND - OR logic $\equiv$ NAND - NAND

POS form : ⟶ $y = (A+\bar{B})(A+C)$



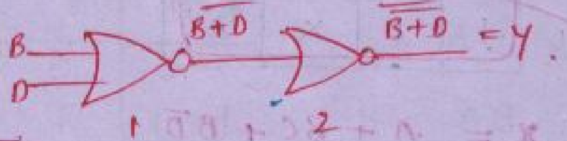
OR – AND logic ≡ NOR – NOR.

Q. How many two i/p NOR gates are required to implement the following k-map.

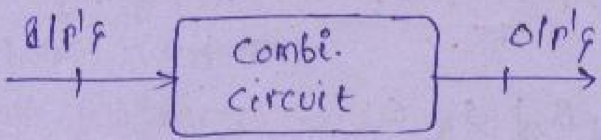
AB \ CD	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	1	1	1	1
10	0	1	1	0

$Y = B + D.$

$\Rightarrow Y = \overline{\overline{B+D}}$

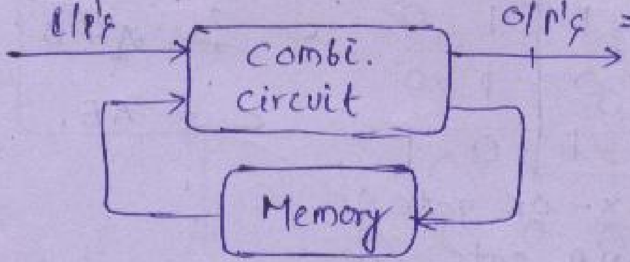


Combinational Circuits :-



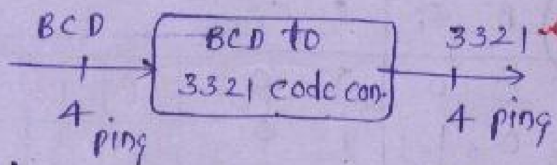
o/p's = f (present i/p's).

Sequential circuits :-



o/p's = f (present i/p's, past o/p's)
(or)
f (present i/p's, present states).

Q. Design a BCD to 3321 code converter.



self complementary

	BCD A	B	C	D	3321 w	x	y	z
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	1
5	0	1	0	1	1	0	1	0
6	0	1	1	0	1	1	0	0
7	0	1	1	1	1	1	0	1
8	1	0	0	0	1	1	1	0
9	1	0	0	1	1	1	1	1

Z

AB \ CD	00	01	11	10
00	0	1	1	0
01	1	0	1	0
11	X	X	X	X
10	0	1	X	X

① $\bar{B}D$ ② CD ③ $B\bar{C}\bar{D}$

$Z = \bar{B}D + CD + B\bar{C}\bar{D}$

y

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	1	0	0
11	X	X	X	X
10	1	1	X	X

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

x

AB \ CD	00	01	11	10
00	0	0	0	0
01	1	0	1	1
11	X	X	X	X
10	1	1	X	X

$x = A + BC + B\bar{D}$

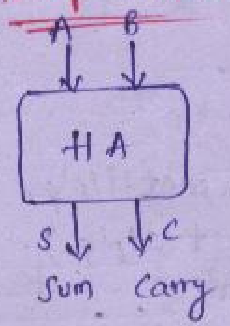
ω

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

$\omega = A + BD + BC$

Arithmetic combi. circuits :-

Half Adder :-



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

$S = A'B + AB'$

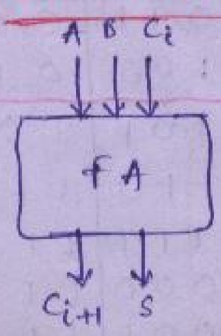
$\Rightarrow A \oplus B$

$C = AB.$

1 HA $\longrightarrow$ { 1 EX-OR gate, 1 AND gate }

* SUNDAY. 12/10/08 *

Full Adder :-



A	B	C_i	S	C_{i+1}
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

for. S

A \ BC_i	00	01	11	10
0	0	1	0	1
1	1	0	1	0

diagonal Adjacency

$$S = \bar{B}(\underbrace{A \oplus C_i}_{x}) + B(\underbrace{A \odot C_i}_{\bar{x}})$$

$$= B \oplus x = B \oplus A \oplus C_i$$

$$\Rightarrow S = A \oplus B \oplus C_i.$$

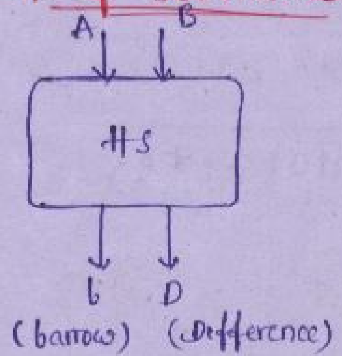
$$C_{i+1} = \bar{A}BC_i + A\bar{B}C_i + AB\bar{C_i} + ABC_i$$

$$= AB + BC_i + C_iA$$

(or)

$$C_i(A \oplus B) + AB.$$

Half Subtractor :-

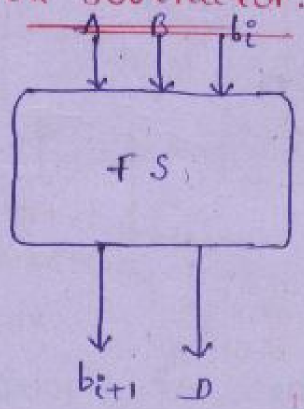


A	B	D	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = A \oplus B$

$b = \bar{A}B.$

full Subtractor :-



A	B	b_i	D	b_{i+1}
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

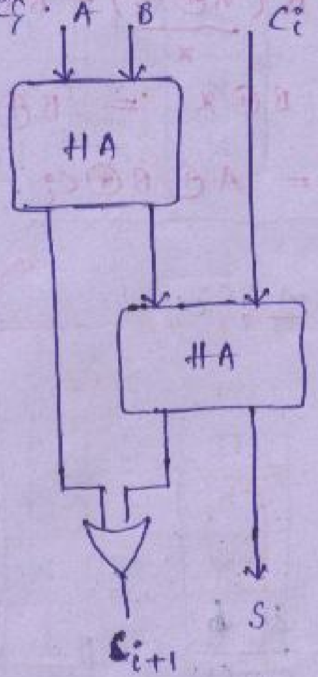
$D = A \oplus B \oplus b_i$

A \ Bb_i	00	01	11	10
0	0	1	1	1
1	0	0	1	0

for b_{i+1}

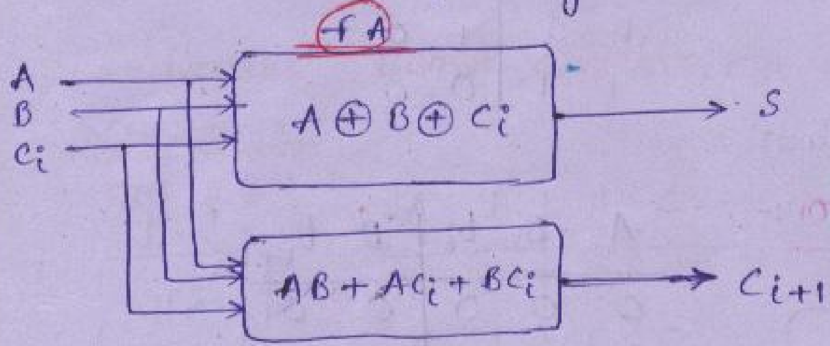
$$\Rightarrow b_{i+1} = \bar{A}b_i + Bb_i + \bar{A}B$$

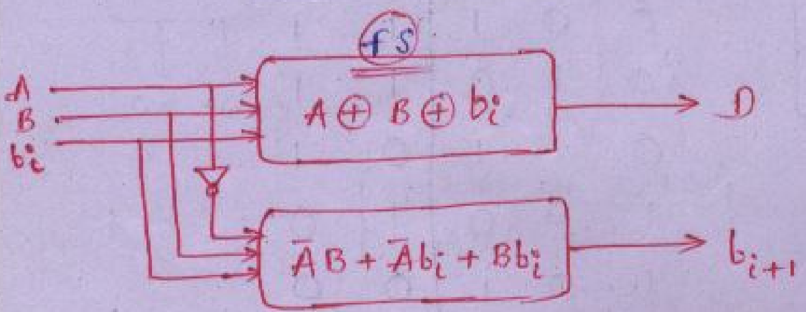
Q. Implement a FA by using HA's and logic gates.



FA requires one OR gate and two HA's.

Q. Convert the following FA into a FS.





Q. Implement the following boolean exp-s using only HA's.

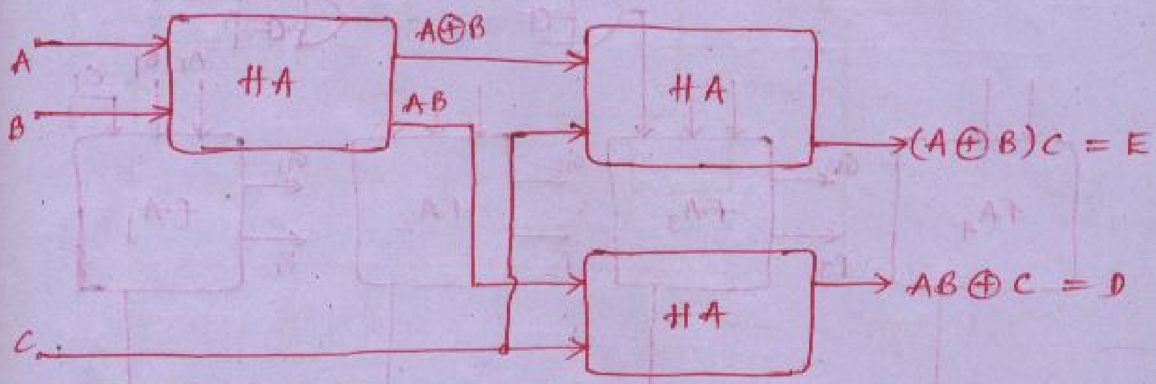
$$D = AB\bar{C} + \bar{A}C + \bar{B}C$$

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

Sol: $D = AB\bar{C} + (\bar{A} + \bar{B})C$

$= AB\bar{C} + \overline{AB} \cdot C = AB \oplus C$

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



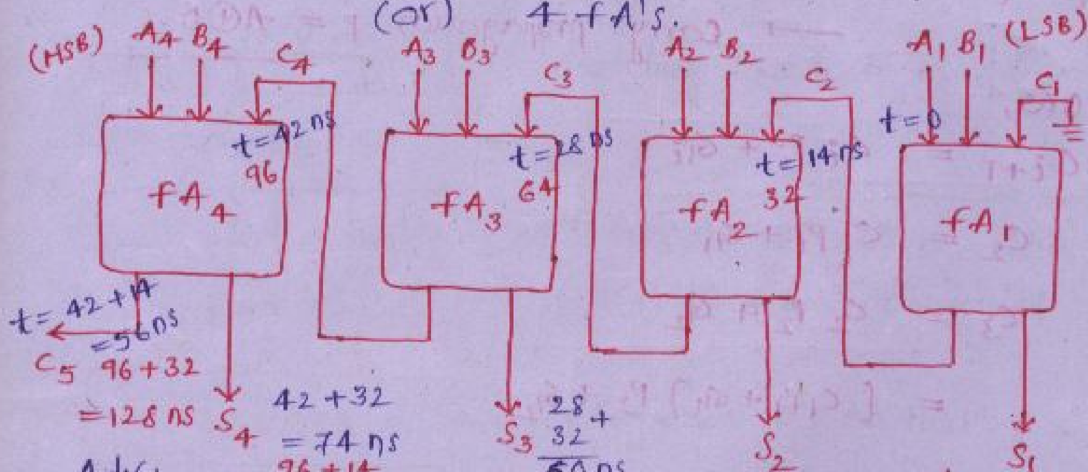
(1). 4-bit parallel Binary Adder :-

$A \longrightarrow \quad A_4 \quad A_3 \quad A_2 \quad A_1$ ↑ (combi. circuit)

$B \longrightarrow \quad B_4 \quad B_3 \quad B_2 \quad B_1$

Required : 3 FA's + 1 HA

(or) 4 FA's.



Adv:

Simple to construct

If $\left\{ \begin{array}{l} 14\text{ ns} \rightarrow \text{sum} \\ 32\text{ ns} \rightarrow \text{carry} \end{array} \right\}$

Drawback:

Total time = 128 ns

Speed of operation is less if the size of the adder increases.

$FA \longrightarrow \left\{ \begin{array}{l} 32\text{ ns} \rightarrow \text{Sum} \\ 14\text{ ns} \rightarrow \text{carry} \end{array} \right\}$ Then the total time required for the operations

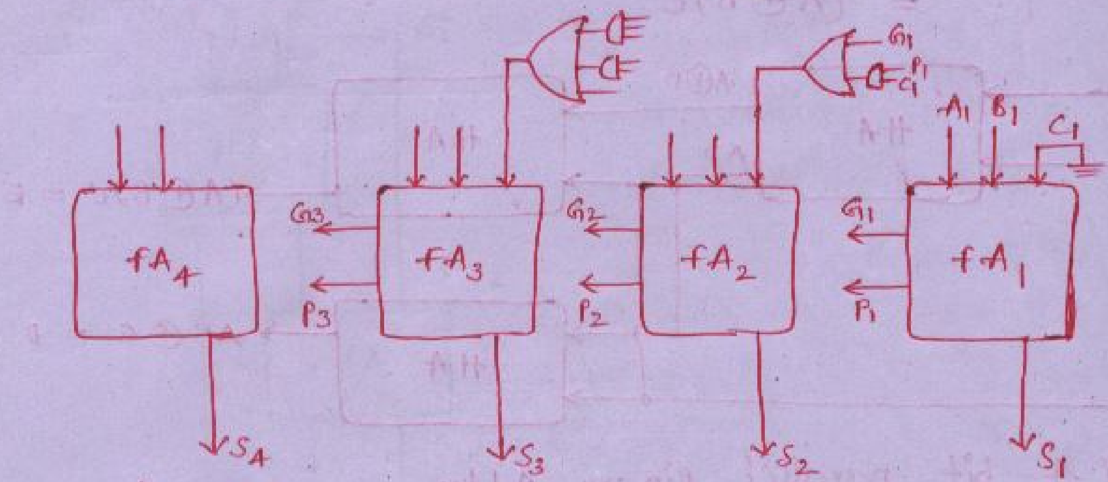
Ans : $42 + 32$ for S_4 ; for $C_5 \Rightarrow 42 + 14 = 56$

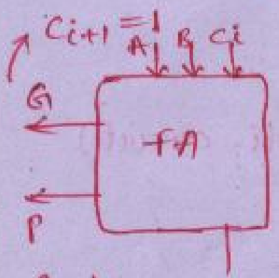
$= 74$ ns.

$\therefore$ To complete addition

Total time = 74 ns.

(2). Carry Look Ahead Adder :- (combi. circuit)





$P=1$

$C_{i+1}=C_i$

$C_{i+1} = C_i(A \oplus B) + AB$.

(1). $C_{i+1}=1$ if $AB=1$

$\longrightarrow$ Carry Generation $G = AB$.

(2). when $C_{i+1} = C_i$ then $A \oplus B = 1$

$\longrightarrow$ Carry propagation $P = A \oplus B$.

Now,

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

$\Rightarrow C_2 = C_1 P_1 + G_1$

$C_3 = C_2 P_2 + G_2$

$= [C_1 P_1 + G_1] P_2 + G_2$

$= C_1 P_1 P_2 + G_1 P_2 + G_2$

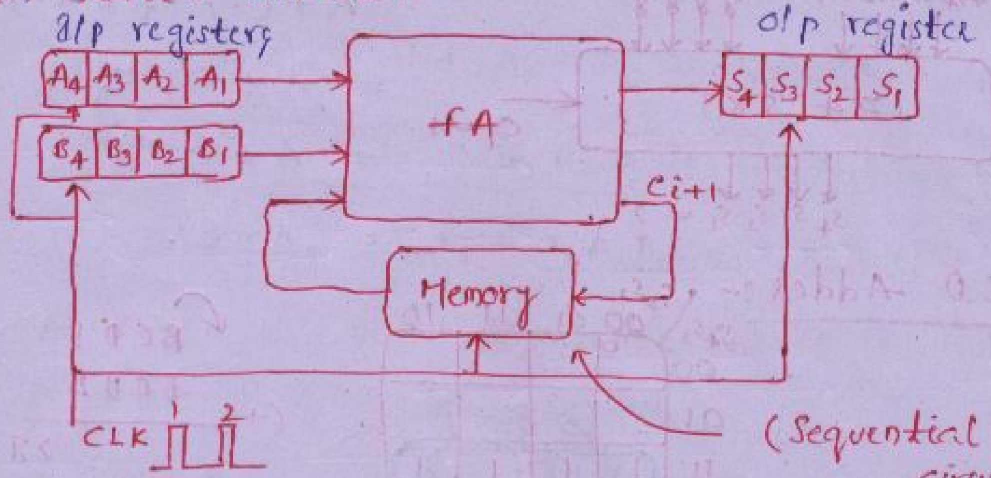
$C_4 = C_3 P_3 + G_3$

$= C_1 P_1 P_2 P_3 + G_1 P_2 P_3 + G_2 P_3 + G_3$

Adv : speed is more

Dis Adv : More hardware complexity

(3). Seriel Adder :-



(Sequential circuit)

Adv: (1) easy to construct

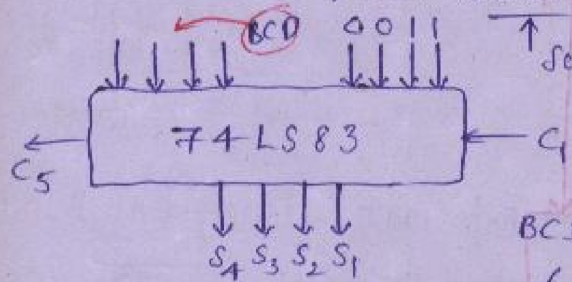
(2). only one FA is used.

Dis-Adv:

speed of operation is less.

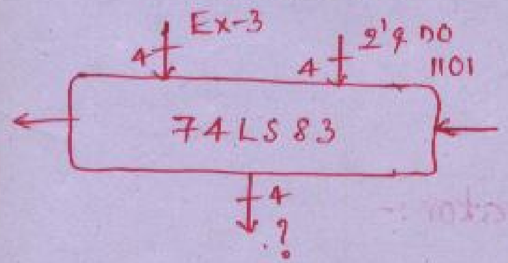
4 Bit parallel Binary Adder (74 LS 83)

LS → Low power Schottky
↑ Switching speed is more.



BCD + 0011 = Ex-3 code

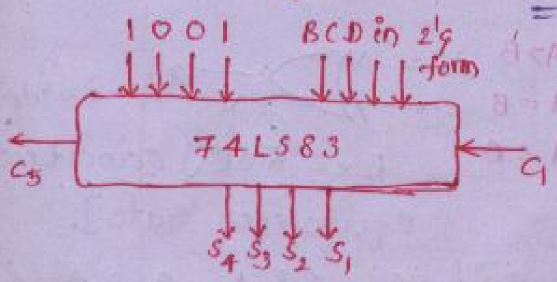
(BCD to Ex-3 converter)



Ex-3 + 2's no 1101

= Ex-3 + (−0011)

= BCD code

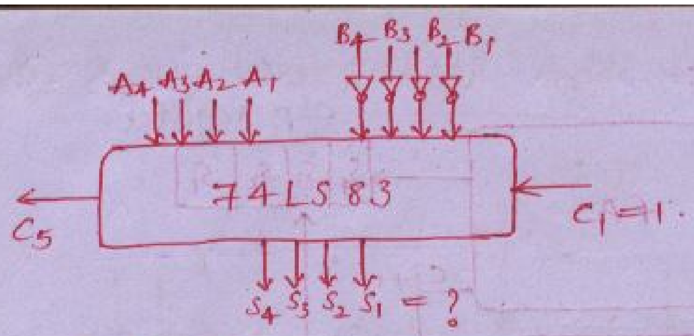


1001 + BCD in 2's comp. form

= 1001 + (− BCD)

= 9's comp. of BCD.

(− x = 2's comp of + x.)



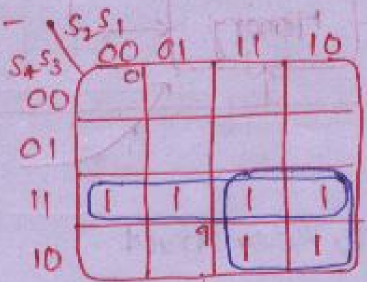
$A + (1\text{'s of } B + 1)$

$= A + 2\text{'s of } B$

$= A + (-B)$

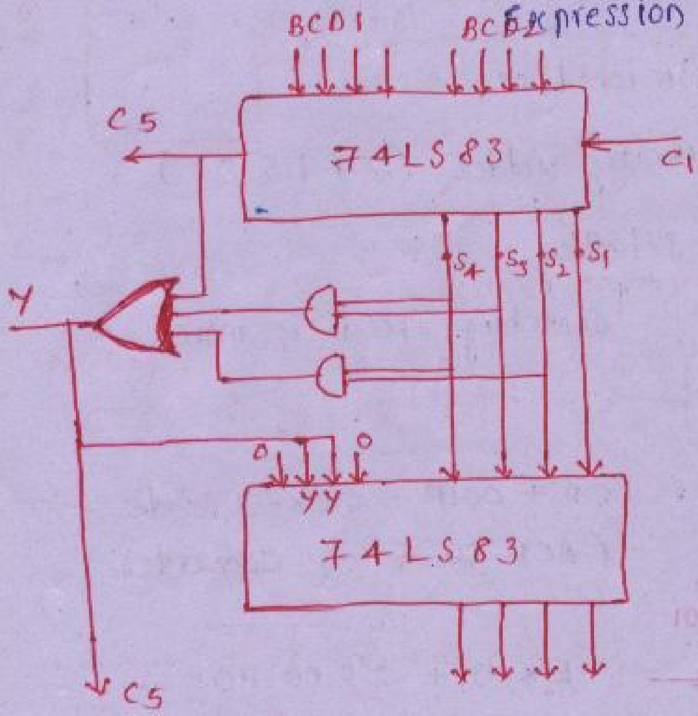
$= A - B$.

BCD Adder:-

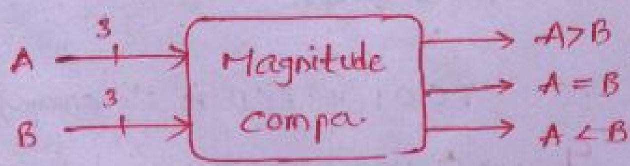


BCD 1
(+) BCD 2
> 9
→ add 6

Invalid BCD expression = $Y = S_4S_3 + S_4S_2 + C_5$.



3-bit magnitude Comparator:-



x_i; $i = 1,2,3$.

Ex-NOR [coincidence, Equivalence Gate].

$A = A_3\ A_2\ A_1$

$B = B_3\ B_2\ B_1$

(a). $A = B$ if $A_3 = B_3$ & $A_2 = B_2$ & $A_1 = B_1$

ie $A = B$ if $x_3 = 1$ & $x_2 = 1$ & $x_1 = 1$

ie $A = B$ if $x_3\, x_2\, x_1 = 1$

(b). $A > B$ if $A_3 > B_3$ (or) $A_3 = B_3$ and $A_2 > B_2$

(or) $A_3 = B_3$ and $A_2 = B_2$ and $A_1 > B_1$

$A > B$ if $A_3\bar{B}_3 + x_3 A_2 \bar{B}_2 + x_3 x_2 A_1 \bar{B}_1 = 1$.

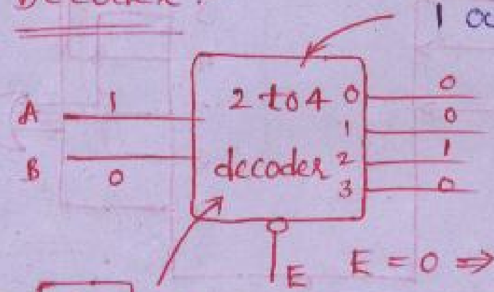
(c). $A < B$ if $\bar{A}_3 B_3 + x_3 \bar{A}_2 B_2 + x_3 x_2 \bar{A}_1 B_1 = 1$.

(1). decoder (2). demultiplexer (3). Encoder (4). Multiplexer

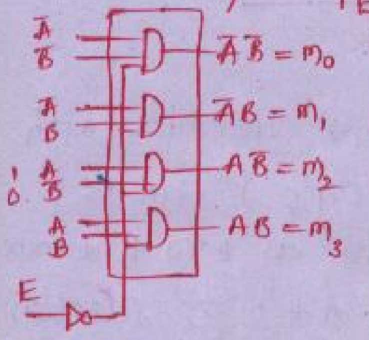
Decoder :-

1 out of 4 decoder

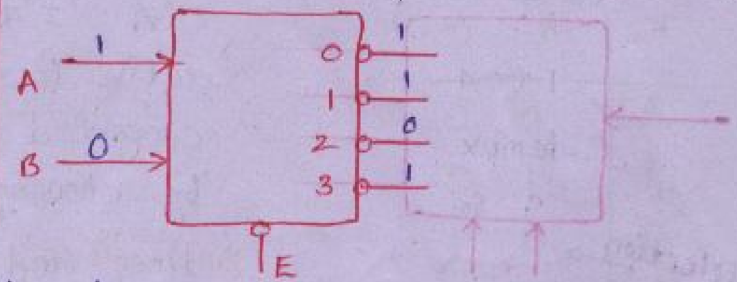
[only one o/p is active out of 4 at any instant].



E = 0 ⇒ decoder is enabled.

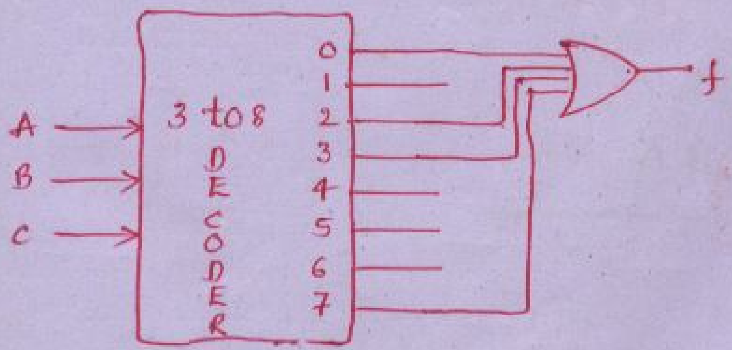


2 to 4 decoder with active low o/p's

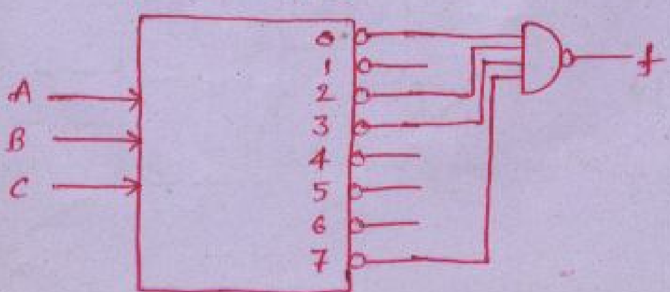


Q. Implement the following sum of minterm eq by using a decoder and logic gates.

$f(A,B,C) = \Sigma m(0, 2, 3, 7)$



	A	B	C	f
m_0	0	0	0	1
	0	0	1	0
m_2	0	1	0	1
m_3	0	1	1	1
	1	0	0	0
	1	0	1	0
	1	1	0	0
m_7	1	1	1	1

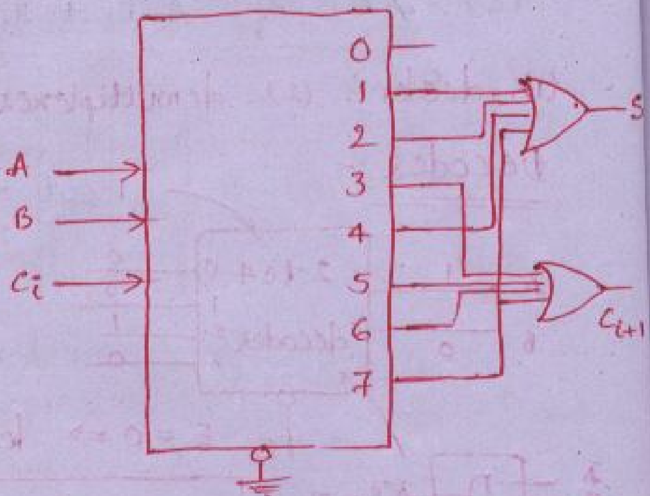


Q. Implement a FA by using decoder and logic Gates.

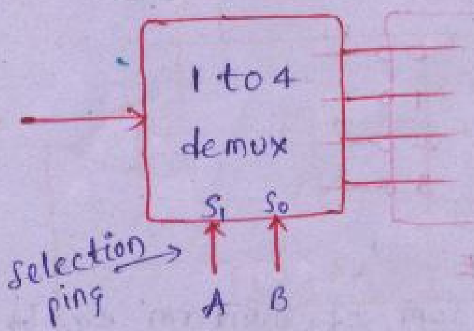
A	B	C_i	C_{i+1}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$S = \Sigma m(1,2,4,7)$

$C_{i+1} = \Sigma m(3,5,6,7)$



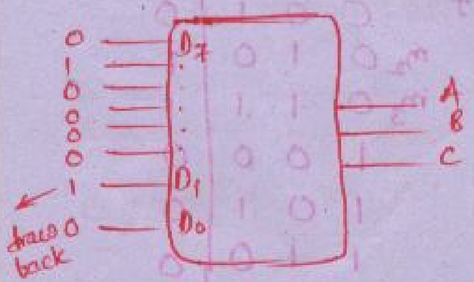
Demultiplexer :-



A 2 to 4 decoder [with active low o/p's] can be converted to a 1 to 4 demux by choosing A & B as selection lines and the enable pin as the serial i/p.

* 29/11/08 *

ENCODER :



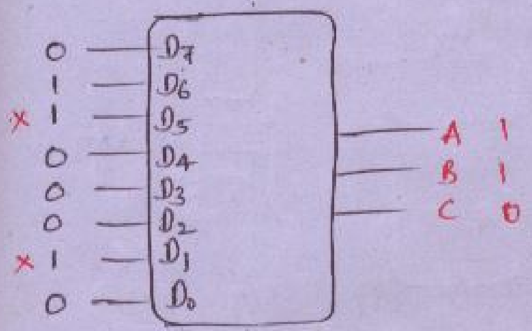
$D_1 \rightarrow 001$

$D_6 \rightarrow 110$

PRIORITY ENCODER: (74 LS 148)

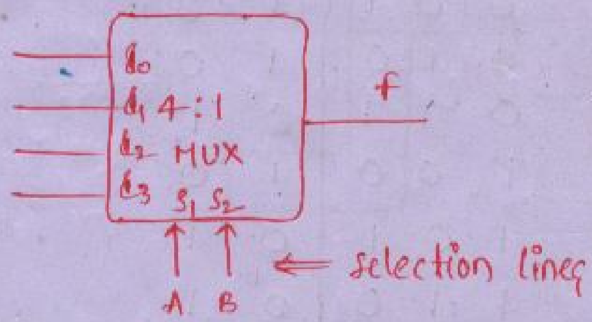
D_7 – highest priority

D_0 – lowest priority



x – ignored

MULTIPLEXER:

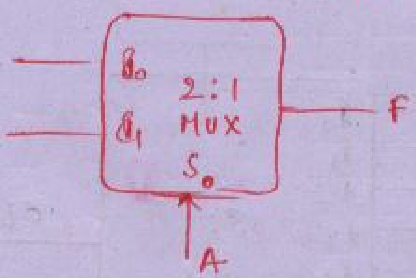


A	B	f
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

for 4:1 MUX,

$$f = \bar{A}\bar{B} I_0 + \bar{A}B I_1 + A\bar{B} I_2 + AB I_3$$

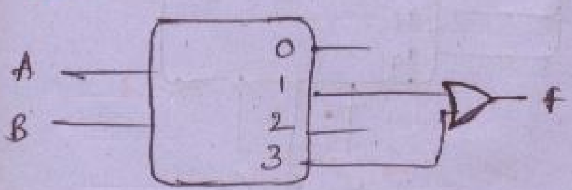
$$= m_0 I_0 + m_1 I_1 + m_2 I_2 + m_3 I_3 .$$



for 2:1 MUX,

$$f = \bar{A} I_0 + A I_1$$

Q.

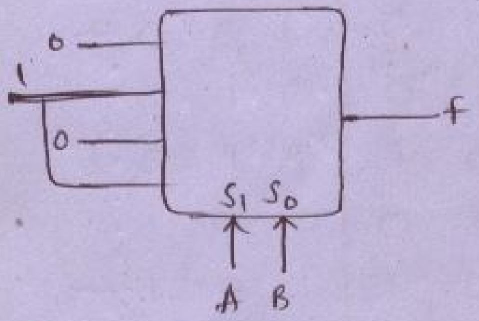


$F(A,B) = \Sigma m(1,3)$

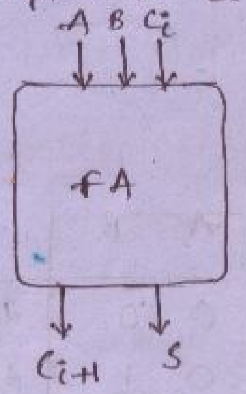
↖ decoder.

Q. Implement the following sominterms exp. by using multiplexer. (sum of minterms)

$f(A,B) = \Sigma m(1,3)$.



Q. Implement a FA by using multiplexers :

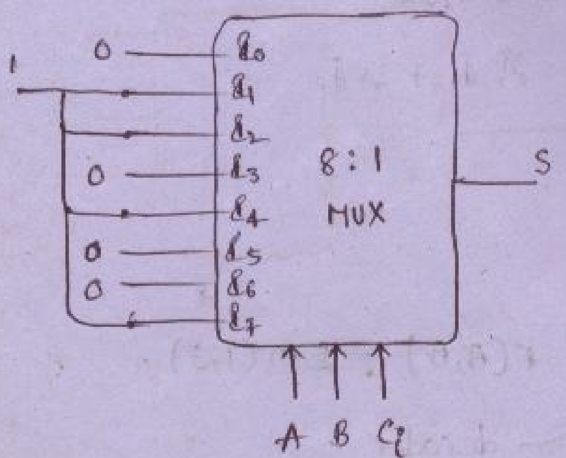


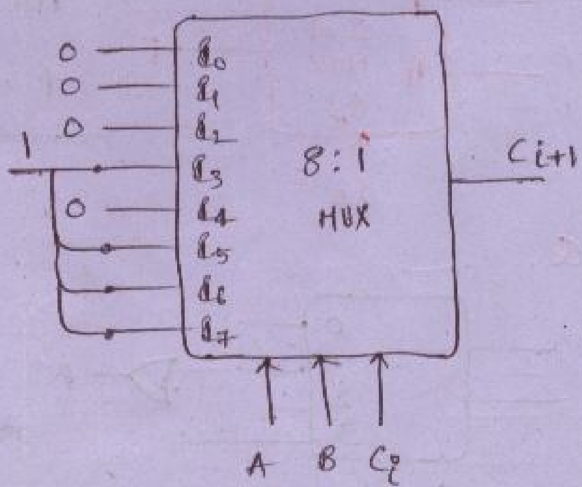
A	B	C_i	S	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

FA ⟶ MUX (sum of minterms)

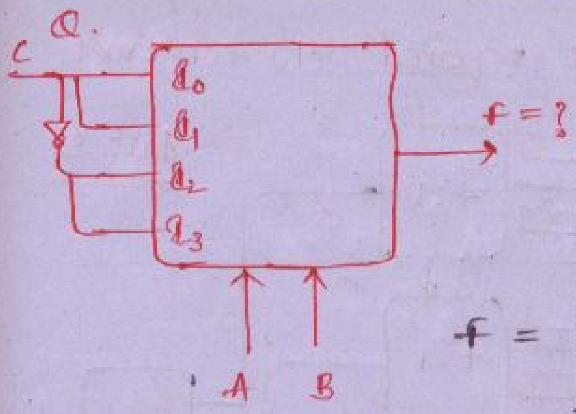
$S = \Sigma m(1,2,4,7)$

$C_{i+1} = \Sigma m(3,5,6,7)$.





Q.



Given that

$I_0 = I_1 = C$

$I_2 = I_3 = \bar{C}$.

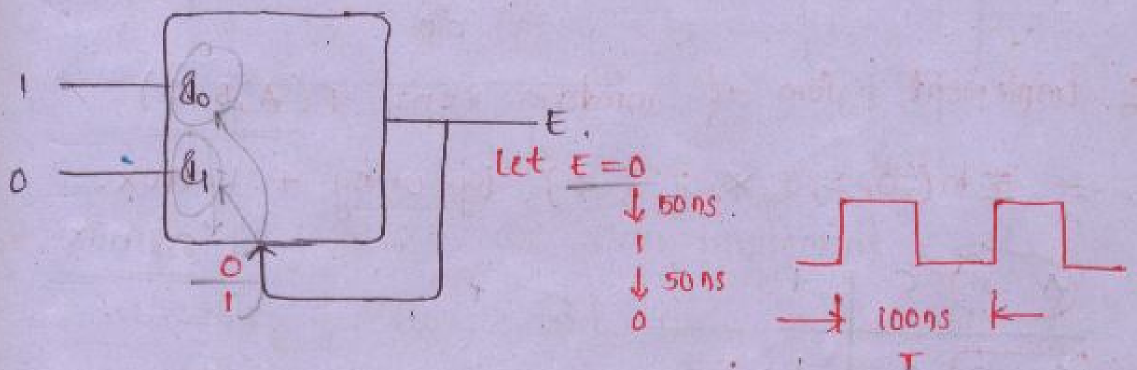
$$f = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$$

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

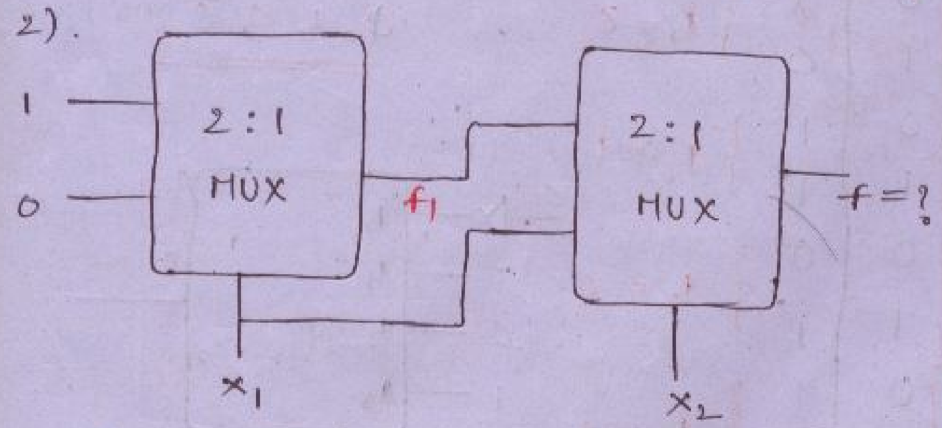
$$= A \oplus C.$$

Q. Determine the o/p's of the following MUX's ?

1). Switching speed is 50 ns.



2).



$$f_1 = \bar{A}I_0 + AI_1$$

$$f_1 = \bar{x}_1 \cdot 1 + x_1 \cdot 0 = \bar{x}_1$$

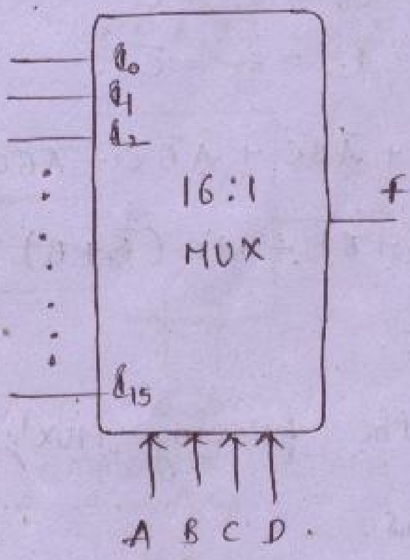
$$f = \bar{A}I_0 + AI_1$$

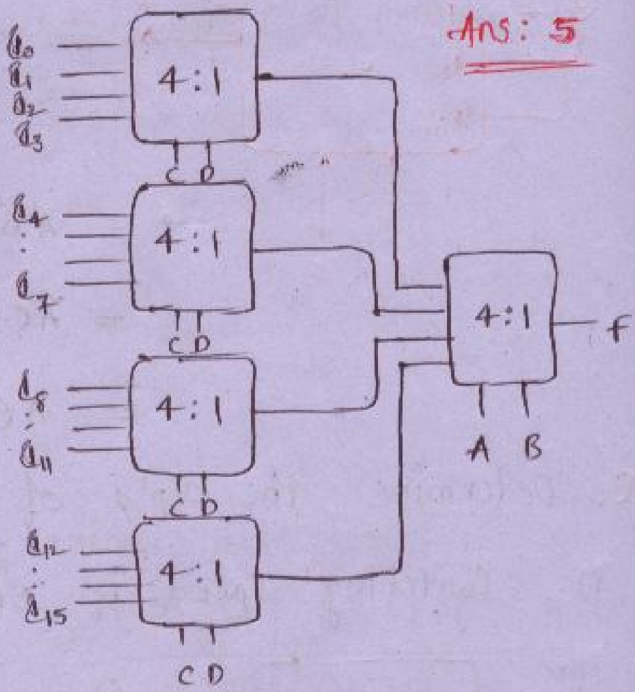
$$= \bar{x}_2 \cdot \bar{x}_1 + x_2 \cdot x_1$$

$$= x_2 \odot x_1$$

Q. How many 4:1 mux's are required to construct a 16:1 MUX.

Ans: 5



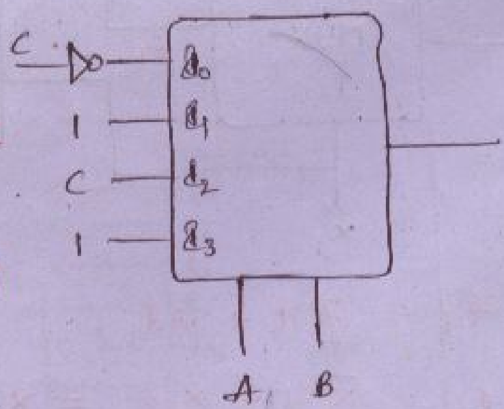


Q. Implement sum of minterm exp. $f(A,B,C) = \Sigma m(0, 2, 3, *, 5, 6, 7)$. by using 4:1 MUX.

8:1 MUX

A	B	C	f	
0	0	0	1	$I_0 = \bar{C}$
0	0	1	0	
0	1	0	1	$I_1 = 1$
0	1	1	1	
1	0	0	0	$I_2 = C$
1	0	1	1	
1	1	0	1	$I_3 = 1$
1	1	1	1	

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



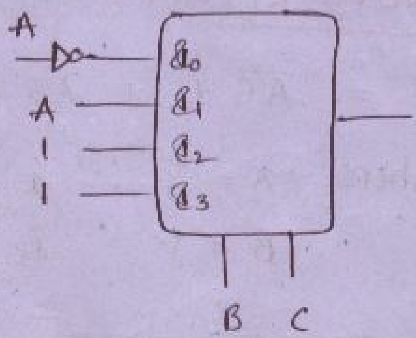
[OR]

AB	00	01	10	11
	I_0	I_1	I_2	I_3
0 $\bar{C}$	000 0	010 2	100 4	6
1 C	001 1	3	5	7

AB	I_0	I_1	I_2	I_3
$\bar{C}$	0	2	4	6
C	1	3	5	7
	$\bar{C}$	1	C	1

Implement above problem by choosing B & C as selection lines.

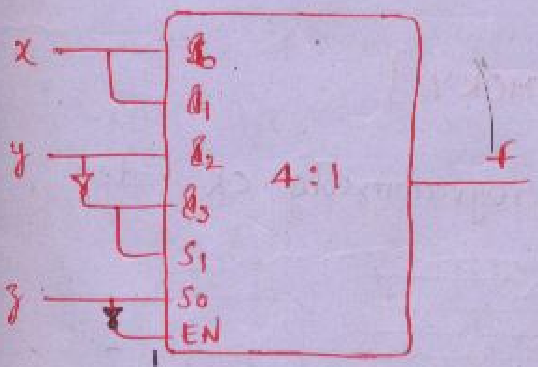
BC	I_0 00	I_1 01	I_2 10	I_3 11
0 $\bar{A}$	0	1	4	6
1 A	4	5	6	7
	$\bar{A}$	A	1	1



* Using 4:1 MUX, we can implement all 2 variable functions $f(A,B)$ and some 3 variable functions. $f(A,B,C)$

⇓ Require some logic gates like NOT GATE.

Q.



If $z=0$, MUX is enabled and with $z=1$, MUX is disabled.

$S_1 = \bar{y}$

$S_0 = z$

x	y	z	S_1 ($\bar{y}$)	S_0 (z)	f
0	0	0	1	0	$I_2 = y = 0$
0	1	0	0	0	$I_0 = x = 0$
1	0	0	1	0	$I_2 = y = 0$
1	1	0	0	0	$I_0 = x = 1$

1 → disabled

$f = xy\bar{z}$

ANOTHER WAY :

$$f = \bar{A}\bar{B} I_0 + \bar{A} B I_1 + A\bar{B} I_2 + AB I_3$$

where S_1 $A = \bar{y}$ $\quad I_0 = I_1 = x$

S_0 $B = z$ $\quad I_2 = y;\ I_3 = \bar{y}$.

$$\Rightarrow f = \bar{\bar{y}}\bar{z}x + \bar{\bar{y}}zx + \bar{y}\bar{z}y + \bar{y}z\cdot\bar{y}$$

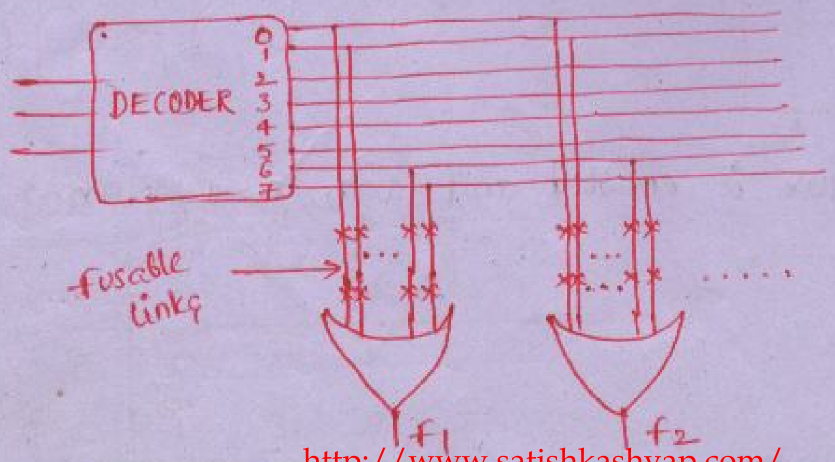
$$= xy\bar{z} + \cancel{xyz} + 0 + \cancel{\bar{y}z}$$

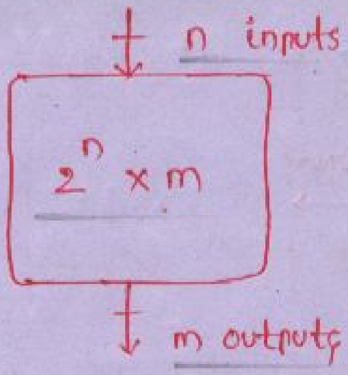
$x=1$, $y=1$, $z=0$ $\qquad$ $x=1$, $y=1$, $z=1$ $\qquad$ $y=0$, $z=1$

$$\Rightarrow f = xy\bar{z}$$

ROM [READ ONLY MEMORY]

ROM ⇒ DECODER + Programmable OR gates





Size of the ROM indicates the no. of FUSES at the beginning

PLA : programmable AND gates & programmable OR gates.

PAL : Programmable AND gates & fixed OR gates.

Decoder
MUX
ROM
} ← Sum of minterms ie $\Sigma m(\ldots)$.
(canonical SOP form).

PLA ← Std. SOP form. is sufficient.

Determine the size of the ROM for the following

(i). $f_1(x,y,z) = \Sigma m(0,1,3)$.

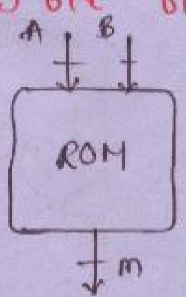
$f_2(x,y,z) = x\bar{y} + \bar{x}\bar{y}\bar{z} + \bar{y}z$

$f_3(x,y,z) = \bar{x}yz$.

$n = 3$ & $m = 3$.

ROM size $= 2^3 \times 3 = 24$.

(ii). 3 bit binary Multiplier



111×111

$7_{10} \times 7_{10} = 49_{10}$

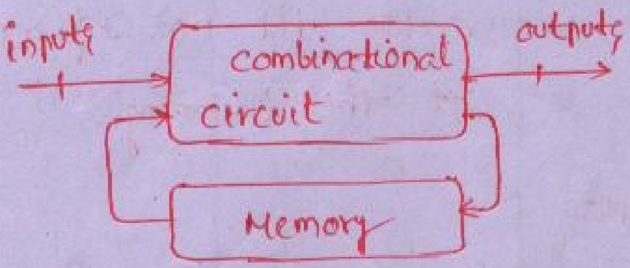
$\Rightarrow 2^m \geq 49$

$\Rightarrow m = 6$.

$\therefore$ Size $= 2^6 \times 6$

$=$

SEQUENTIAL CIRCUITS:



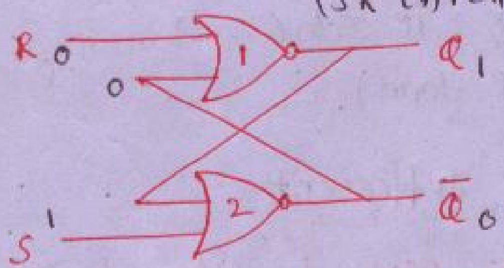
outputs = f (present i/p's, past o/p's)

or

f (" , present state).

1 Bit Memory Element:

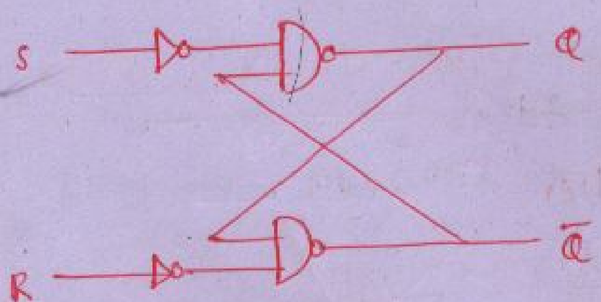
(SR LATCH)



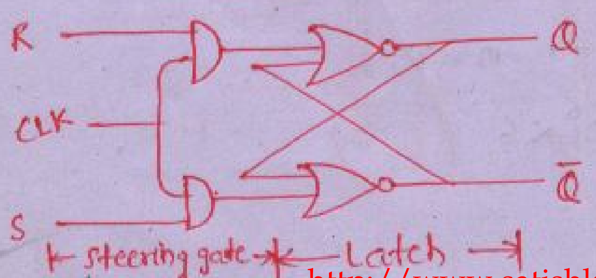
S (SET)	R (RESET)	Q
0	0	No change in o/p
0	1	0
1	0	1
1	1	impractical state

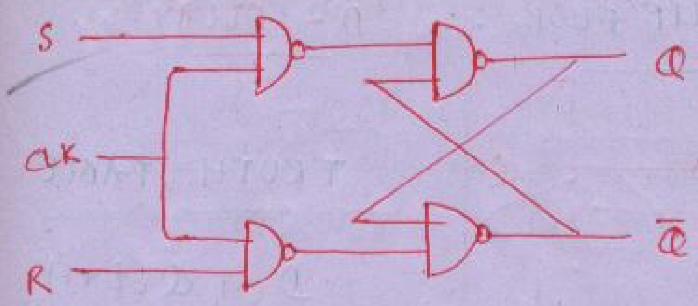
S R	Q
1 0	1
0 0	1

→ Even i/p's are removed the o/p will be 1 ie it stored the o/p. → 1 bit memory unit



CLOCKED S-R FLIP FLOP:

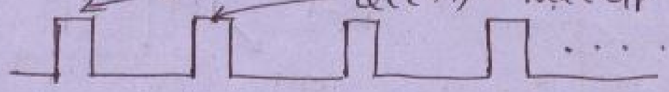




|← Steering gate →|← Latch →|

$Q(t)$ = present o/p

$Q(t+1)$ = Next o/p



CLK waveform.

TRUTH TABLE :

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	(Ambiguous State)

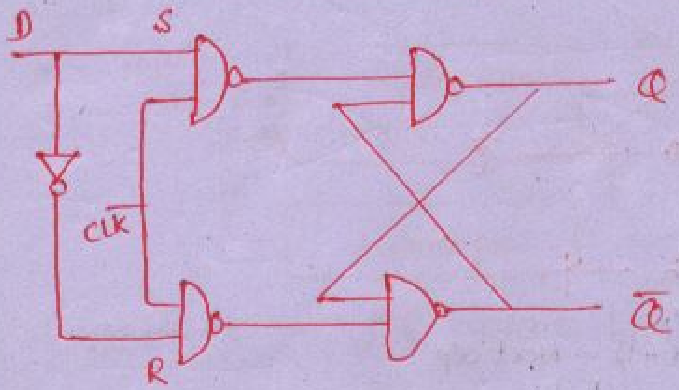
CHAR. TABLE :

S	R	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

S \ RQ	00	01	11	10
0	0	1	0	0
1	1	1	X	X

$$Q(t+1) = S + \bar{R}Q.$$

CLOCKED D- FLIP FLOP : D - DELAY



TRUTH TABLE

D	Q(t+1)
0	0
1	1

S=0
R=1

S=1
R=0

CHAR. TABLE :

D	Q(t)	Q(t+1)
0	0	0
0	1	0
1	0	1
1	1	1

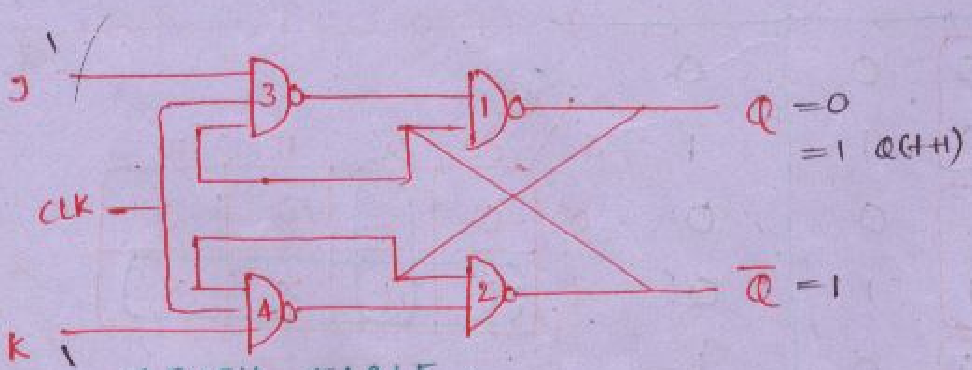
$Q(t+1) = D\bar{Q} + DQ$

$= D.$

CLOCKED JK FLIP FLOP :

$S = J\bar{Q}$

$R = KQ$



TRUTH TABLE :

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

CHAR. TABLE:

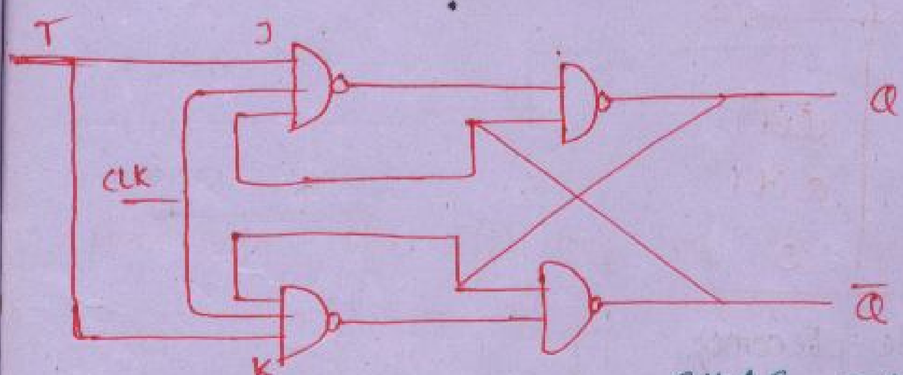
J	K	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

J \ KQ	00	01	11	10
0	0	1	0	0
1	1	1	0	1

$$Q(t+1) = J\bar{Q} + \bar{K}Q$$

CLOCKED T- FLIP FLOP:

T - TOGGLE



TRUTH TABLE :-

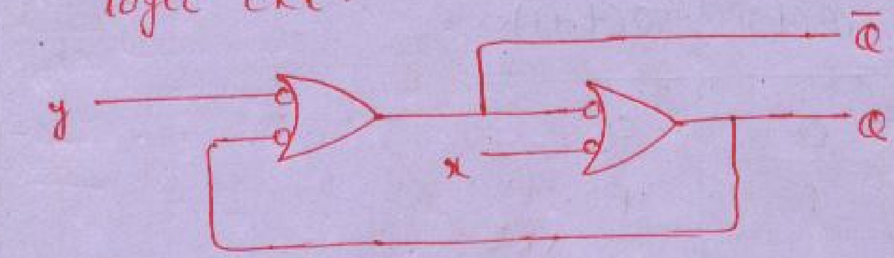
T	Q(t+1)
J=K=0 ← 0	Q(t)
J=K=1 ← 1	$\overline{Q(t)}$

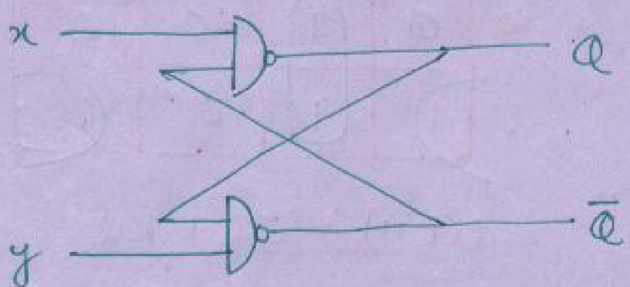
CHAR. TABLE :-

T	Q(t)	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

$$\therefore Q(t+1) = T \oplus Q$$

Q. Determine the fun. table of the following logic ckt.





x	y	Q
0	0	$Q=1, \bar{Q}=1$ —
0	1	1
1	0	0
1	1	No change

Q. obtain char. eq. of x-y flip flop whose truth table as shown below —

x	y	Q(t+1)
0	0	1
0	1	$\overline{Q(t)}$
1	0	Q(t)
1	1	0

char. table becomes :-

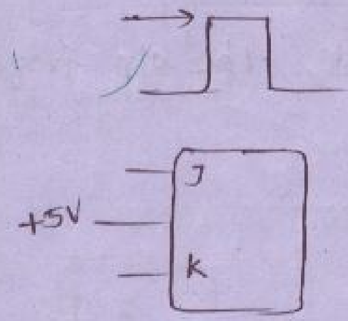
x	y	Q(t)	Q(t+1)
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

x \ yQ	00	01	11	10
0	1	1	0	1
1	0	1	0	0

$$Q(t+1) = \bar{x}\bar{Q} + \bar{y}Q$$

TYPES OF TRIGGERING:

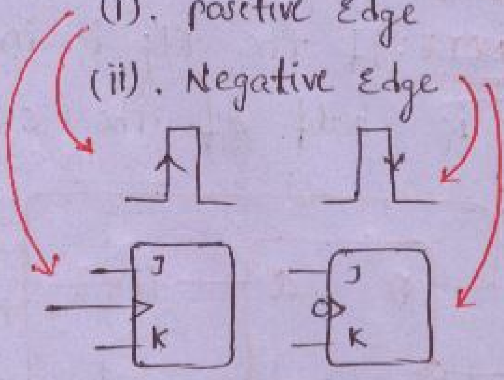
(1). LEVEL TRIGGER

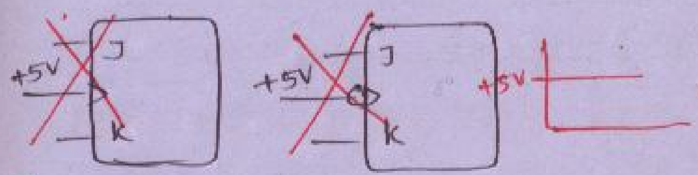


(2). EDGE TRIGGERED

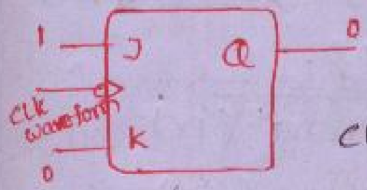
(i). positive Edge

(ii). Negative Edge

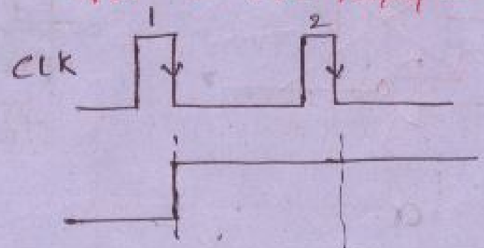




Q.



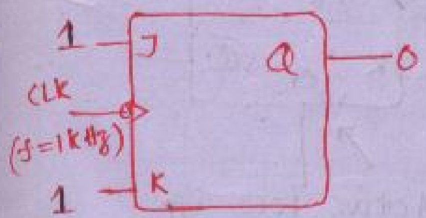
for 2 clk pulses — ?

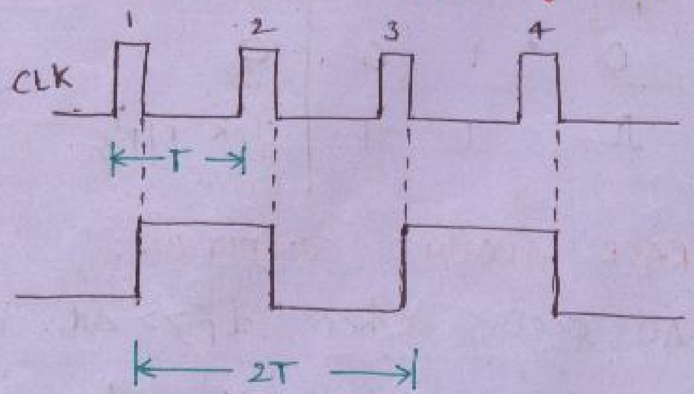


$Q(t) = 0$

$Q(t+1) = 1$

Q. Determine the o/p freq. of the following F/f — ?





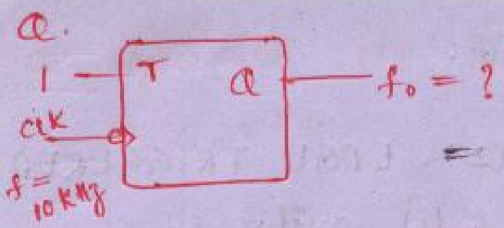
J	K	
1	1	$\rightarrow Q(t+1) = \overline{Q(t)}$

$T_0 = 2T$

$f_0 = \dfrac{1}{T_0}$

$f_0 = \dfrac{f}{2} = \dfrac{1\,kHz}{2} = 500\,Hz$

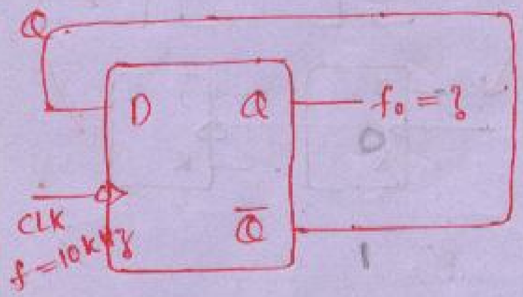
Q.



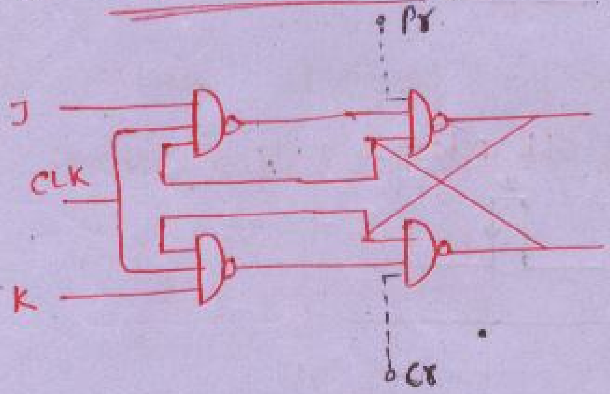
$= \frac{f}{2} = 5$ kHz.

NOTE: If the F/F is in toggle mode, the o/p freq. is half of the clk freq.

Q.



* SUN. 07/12/08 *

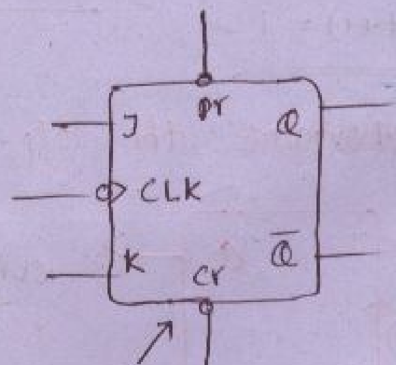


Asynchronous / direct i/p's:

Preset (Pr)

clear (Cr)

CLK	Pr	Cr	Q
0	0	1	1
0	1	0	0
⎍	1	1	J, K i/p's.



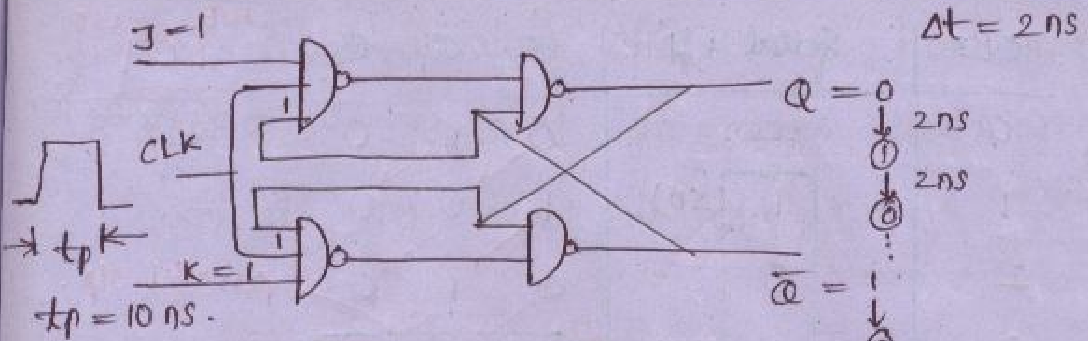
Active low i/p

RACE AROUND CONDITION:

RAC occurs when $t_p >> \Delta t$ and $J = K = 1$.

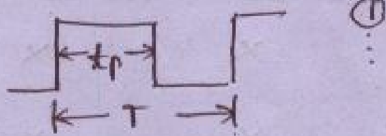
$t_p \Rightarrow$ applied clk pulse width

$\Delta t \rightarrow$ propogation delay of F/F.



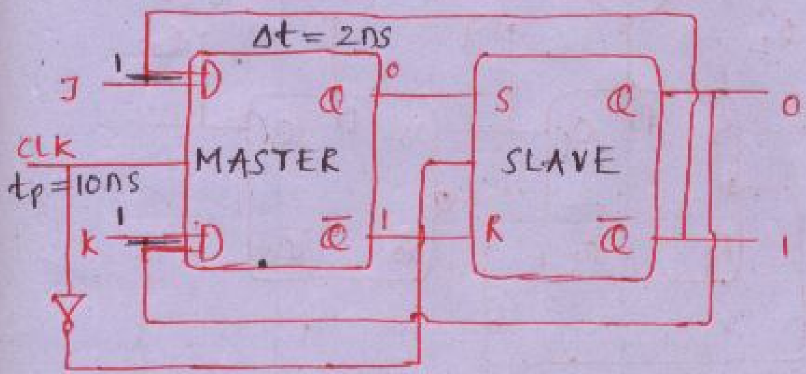
TO AVOID RAC :

$t_p \le \Delta t < T$.

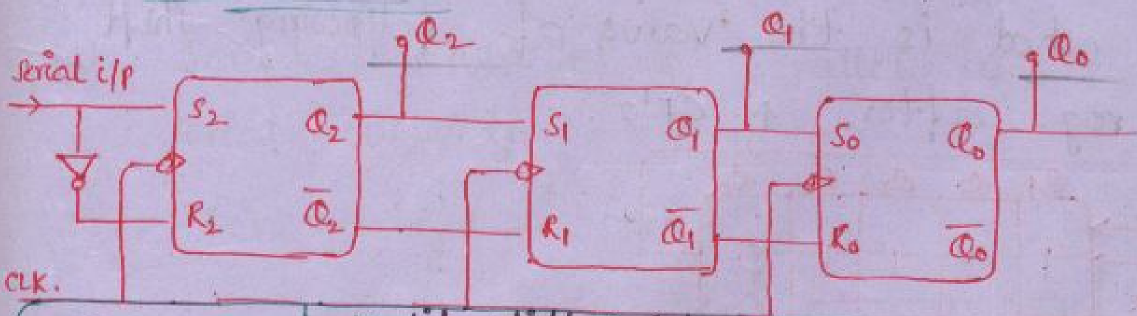


RAC occurs only in level triggered f/f but not in edge triggered f/f.

MASTER - SLAVE JK f/f :

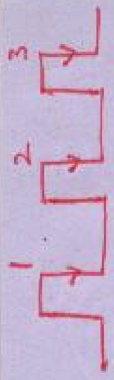


SHIFT REGISTER ⟶ D - f/f's.



3 bit shift REGISTER

	N-bit shift Register
(1). SIPO	'N' CLKS.
(2). SISO	N + (N−1) = (2N−1) clk's
(3). PIPO	NO CLK's.
(4). PISO	(N−1) CP's.

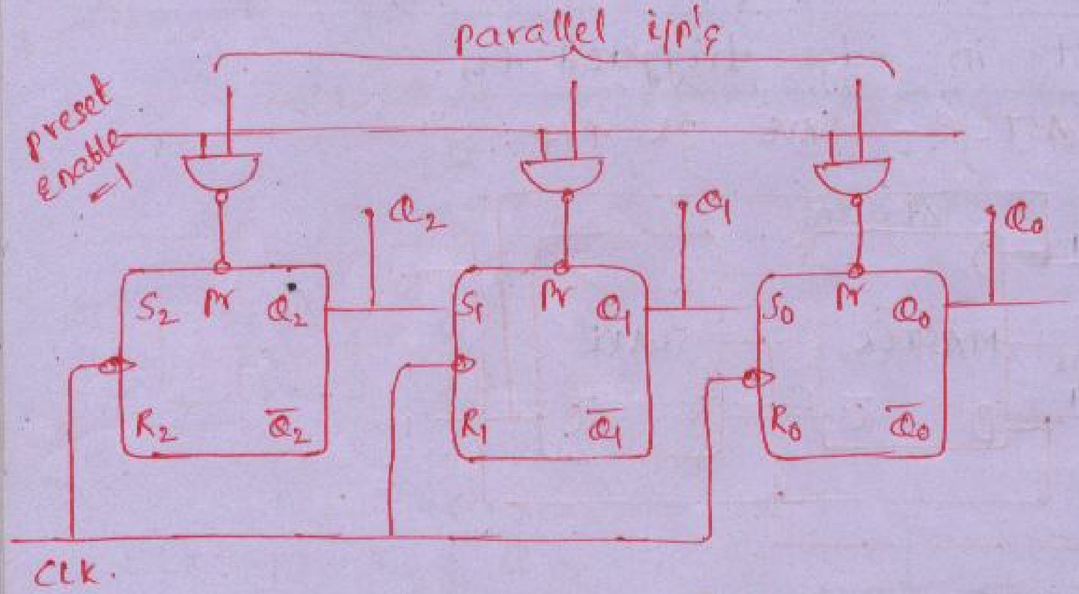


CLK	Serial i/p	Q_2	Q_1	Q_0
0	—	0	0	0
1	1 (LSB)	1	0	0
2	0	0	1	0
3	1 (MSB)	1	0	1
4	x	x	1	0
5	x	x	x	1

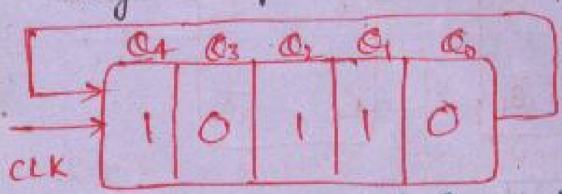
MSB LSB
1 0 1
$(\overline{Q_2 Q_1 Q_0})$

parallel o/p.

→ serial o/p

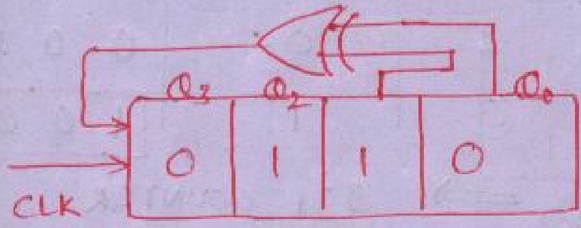


Q. what is the value of following shift reg. after 4 CP's.



CLK	Serial i/p (Q_0)	Q_4	Q_3	Q_2	Q_1	Q_0
0	—	1	0	1	1	0
1	0	0	1	0	1	1
2	1	1	0	1	0	1
3	1	1	1	0	1	0
4	0.	0	1	1	0	1
5	1	[illegible]	0	1	1	0

Q. In the following shift reg. how many CP's are required to make shift reg. content to have all one's.



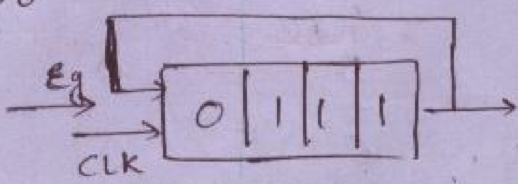
CLK	Serial i/p ($Q_1 \oplus Q_0$)	Q_3	Q_2	Q_1	Q_0
0	—	0	1	1	0
1	1	1	0	1	1
2	0	0	1	0	1
3	1	1	0	1	0
4	1	1	1	0	1
5	1	1	1	1	0
6	1	1	1	1	1

APPLICATIONS OF SHIFT REG'S:

(1). Serial to parallel & parallel to serial convertion

(2). Time delays – SISO

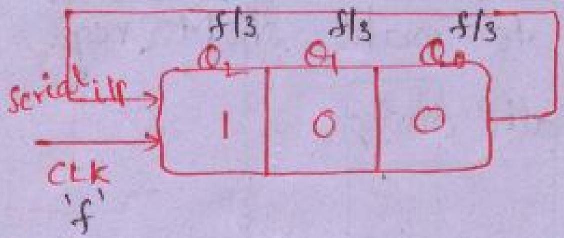
(3). Sequence Generator



.... 0111011101110111.

(4). Counter → RING
→ JOHNSON.

RING COUNTER:

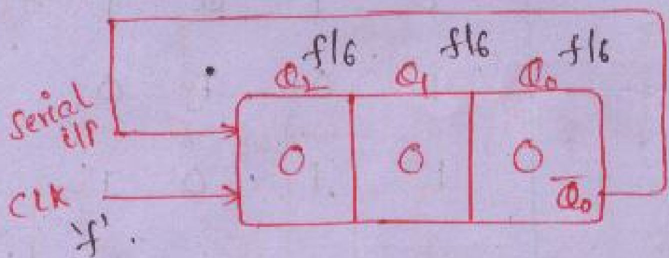


CLK	Serial i/p (Q_0)	Q_2 Q_1 Q_0
0	—	1 0 0
1	0	0 1 0
2	0	0 0 1
3	1	1 0 0

$\Longrightarrow$ 3:1 COUNTER

N- bit Ring Counter:
→ counting capacity = N:1
→ output frequency = f/N.

JOHNSON COUNTER: [TWISTED RING COUNTER].



CLK	Serial i/p ($\bar{Q}_0$)	Q_2	Q_1	Q_0
0	—	0	0	0
1	1	1	0	0
2	1	1	1	0
3	1	1	1	1
4	0	0	1	1
5	0	0	0	1
6	0	0	0	0

$\Longrightarrow$ 6:1 COUNTER.

N - Bit Johnson Counter:
→ Counting capacity = 2N:1
→ Output frequency = f/2N.

Q. what is the o/p freq. of a 3 bit Johnson counter if its clk freq is 18 kHz. The initial content of the reg. is 101.

CLK	Serial i/p ($\bar{Q}_0$)	Q_2	Q_1	Q_0
0	—	1	0	1
1	0	0	1	0
2	1	1	0	1

→ 2:1 COUNTER.

$2N = 6$

$\therefore f_0 = \frac{f}{2} = 9\ kHz.$

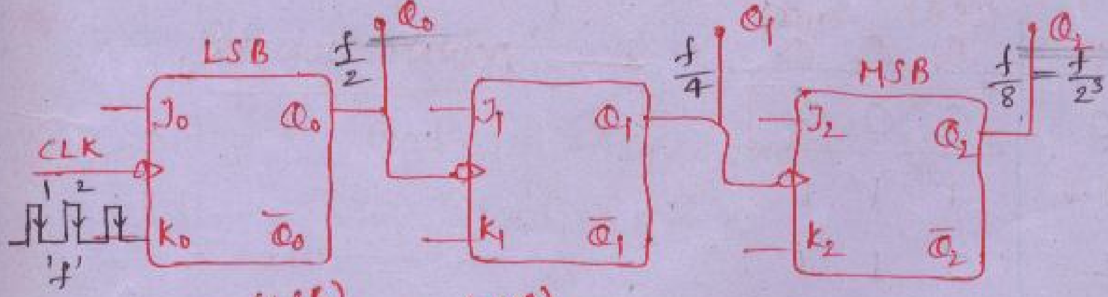
$f/_{2N} = f/6$

COUNTERS:

(1). Asynchronous / Ripple. → T' f/f.

(2). Synchronous / parallel.

3- bit Asynchronous / Ripple counter:



CLK	Q_0 (LSB)	Q_1	Q_2 (MSB)
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	0	1	1
7	1	1	1
8	0	0	0

10 ns
20 ns
30 ns → 8:1 COUNTER

← UP COUNTER

{ CLK PULSE is given to LSB f/f }

→ N - bit Asynchronous counter :

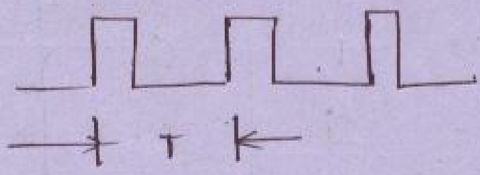
→ $2^N : 1$ counter

→ final o/p freq $= f/2^N$.

Let $t_{pd/FF} = 10$ ns.

Then Max. convertion time = 30 ns.

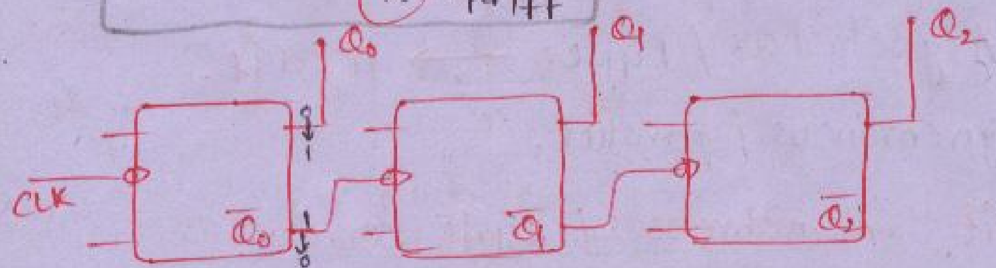
$\Rightarrow T \geq 30$ ns.



$f = \frac{1}{T} \leq \frac{1}{30\text{ ns}}$

$\rightarrow f_{max} = \frac{1}{30\text{ ns}}$.

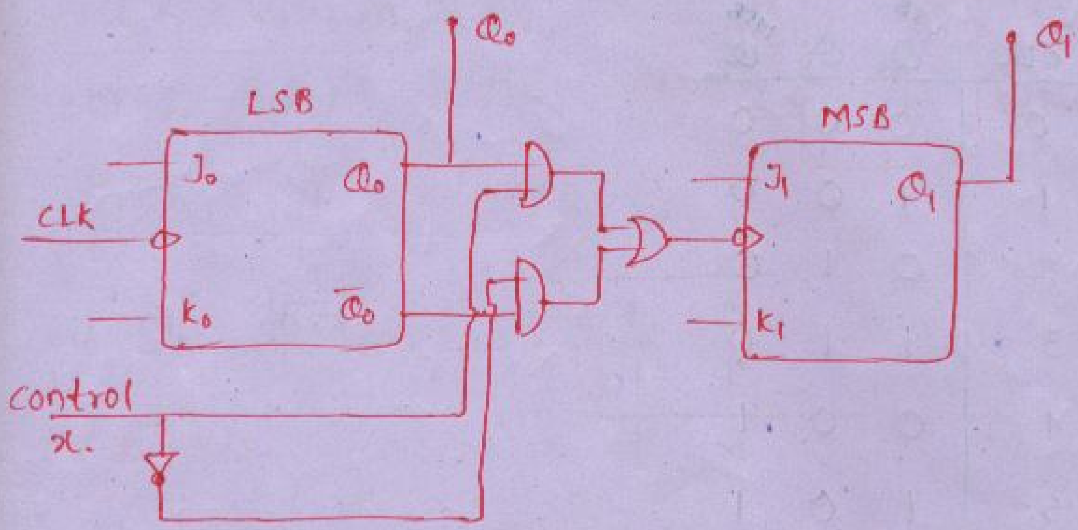
$\therefore \boxed{f_{max} = \frac{1}{N \cdot t_{pd/FF}}}$ no. of f/f's.



CLK	Q_0 (LSB)	Q_1	Q_2 (MSB)
0	0	0	0
1	1	1	1
2	0	1	1
3	1	0	1
4	0	0	1
5	1	1	0
6	0	1	0
7	1	0	D
8	0	0	0

← DOWN COUNTER.

2- Bit Asynchronous up/down counter:



$x = 1, \longrightarrow Q_0 \longrightarrow CLK \longrightarrow$ up counter. (00, 01, 10, 11, 00..

$x = 0, \longrightarrow \bar{Q}_0 \longrightarrow CLK \longrightarrow$ DOWN counter. (00, 11, 10, 01, 00..

MODULUS OF A COUNTER:

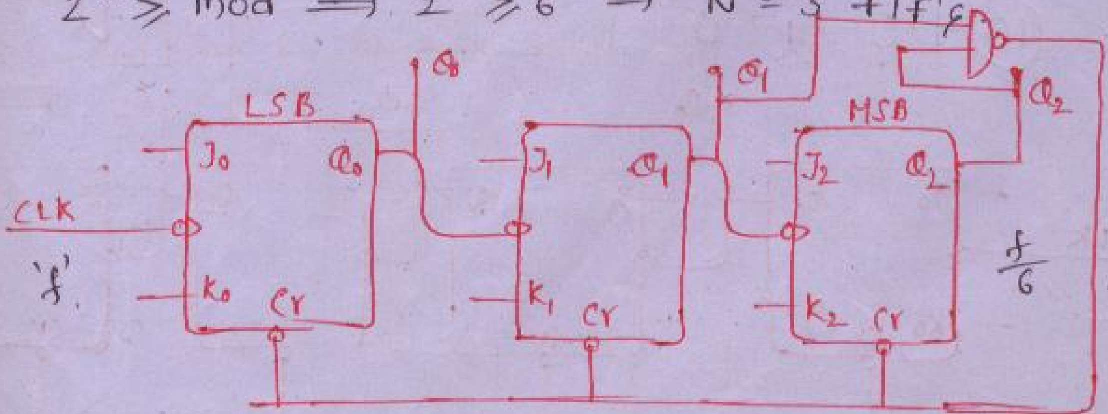
→ It is the no. of CP's required to bring the counter to the initial state.

→ A Mod - N counter counts from 0 to (N-1). and o/p freq. = $\frac{f}{N}$.

Q. Construct Mod - 6 Asy. counter.

MOD - 6 ASY. COUNTER:

$2^N \geq mod \Rightarrow 2^N \geq 6 \Rightarrow N = 3$ F/F's



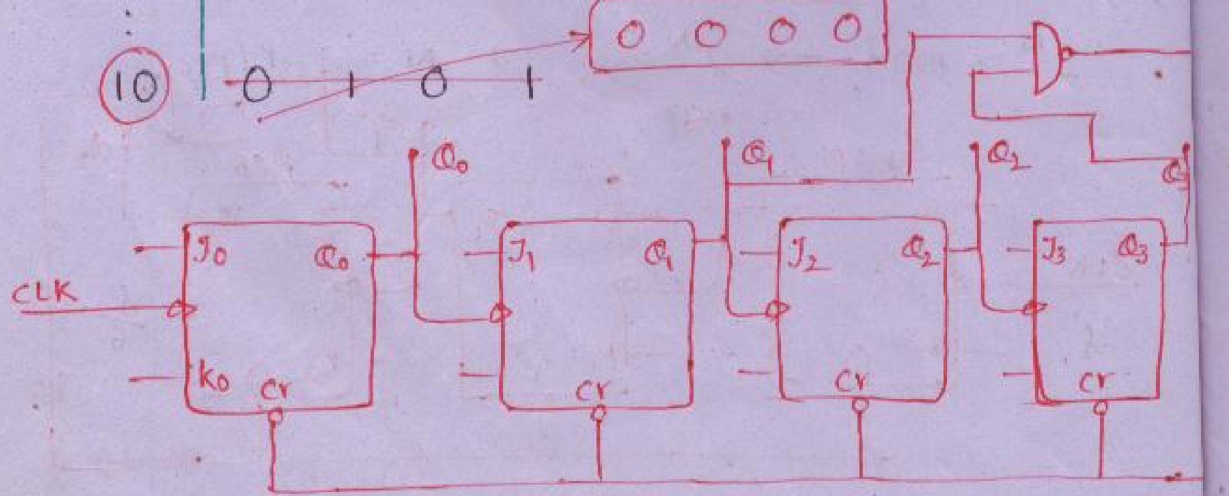
UP COUNTER

CLK	Q_0 (LSB)	Q_1	Q_2 (MSB)
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	~~0~~	~~1~~	~~1~~ → $\boxed{0\ 0\ 0}$
7			

Q. construct a $\underbrace{\text{Asy. decade}}_{\text{Mod}-10}$ counter ?

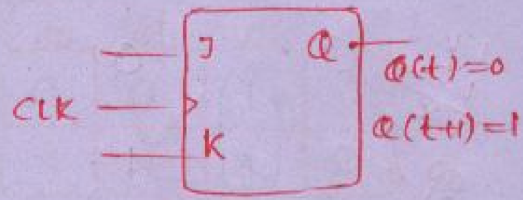
$2^N \geq 10 \longrightarrow N = 4$ f/f's.

CLK	Q_0 (LSB)	Q_1	Q_2	Q_3 (MSB)
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
⋮				
10	~~0~~	~~1~~	~~0~~	~~1~~ → $\boxed{0\ 0\ 0\ 0}$



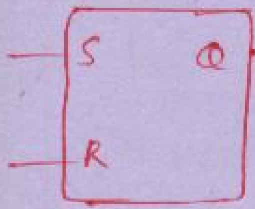
Excitation Tables :

J	K	$Q(t+1)$
~~0~~	~~0~~	$Q(t)$
~~0~~	~~1~~	0
1	0	1
1	1	$\overline{Q(t)}$



J	K
1	0
1	1
1	x .

SR f/f :



S	R
0	1
0	0
0	x .

$Q(t)$	$Q(t+1)$	J	K	S	R	T	D
0	0	0	x	0	x	0	0
0	1	1	x	1	0	1	1
1	0	x	1	0	1	1	0
1	1	x	0	x	0	0	1

Q. Obtain Excitation table of x-y f/f :-

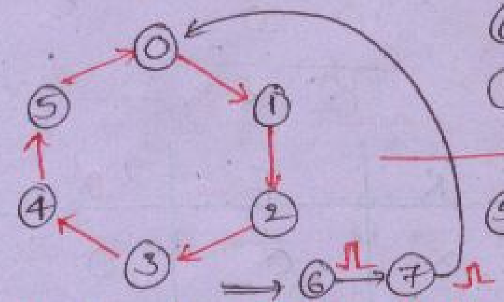
X	Y	$Q(t+1)$
0	0	1
0	1	$\overline{Q(t)}$
1	0	$Q(t)$
1	1	0

	Q(t)	Q(t+1)	x	y
11, 10 → 1X	0	0	1	X
00, 01 → 0X	0	1	0	X
11, 01 → X1	1	0	X	1
00, 10 → X0	1	1	X	0

Q. Design a mod-6 Syn. counter using JK f/f's.

mod-6 ⟶ 0 to 5.

- state diagram



0 — 000

1 — 001

⋮

5 — 101

self starting counter.

state transition Table:

Present state Q_2	Q_1	Q_0	Next state Q_2	Q_1	Q_0	FF inputs J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	0	0	0	X	1	0	X	X	1

$J_0 = K_0 = 1.$

J_1

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	0	1	X	X
1	0	0	X	X

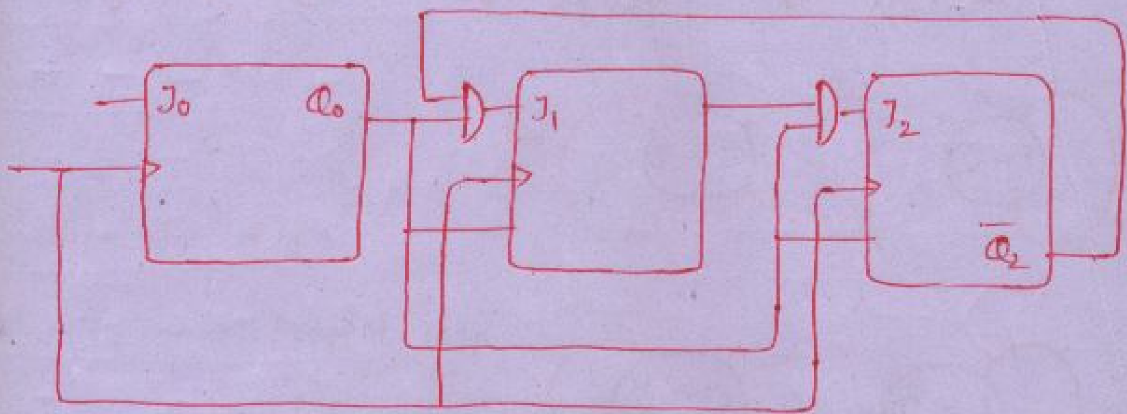
$J_1 = \bar{Q}_2 Q_0$

J_2

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_2 = Q_1 Q_0$

and $K_1 = Q_0$, $K_2 = Q_0$



$$f_{max} = \frac{1}{t_{pd}/ff}$$

Unused States

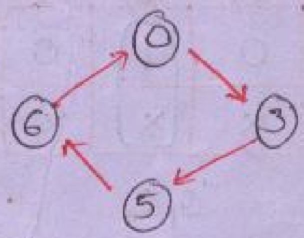
⑥
⑦

$\longrightarrow$ ⑥ $\xrightarrow{\sqcap}$?

Present state			f/f i/p's						Next state		
Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0	Q_2	Q_1	Q_0
1	1	0	0	0	0	0	1	1	1	1	1
1	1	1	1	1	0	1	1	1	0	0	0

Q. Design a syn. counter using T f/f's. which counts to 0, 3, 5, 6, 0, . . .
Is it a self starting counter ?

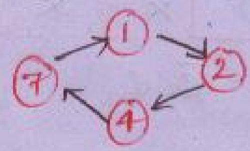
State diagram → Not a self starting counter.



"LOCK OUT"

Unused states:

1, 2, 4, 7



(1).

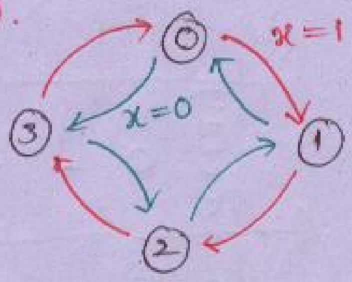
P.S.			N.S			f/f i/p's		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	T_2	T_1	T_0

(2).

P.S	f/f i/p's	N.S.

Q. Draw the state diagram of following digital circuit (i)(2 bit syn. up/down counter).

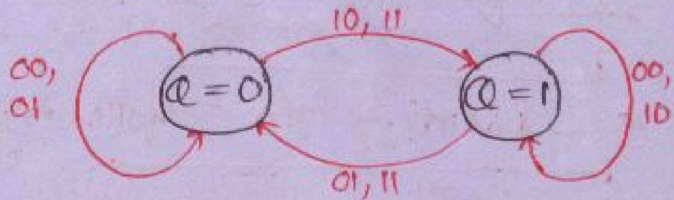
(i).



(ii). JK - f/f

present i/p's { J, K → branches of each state.

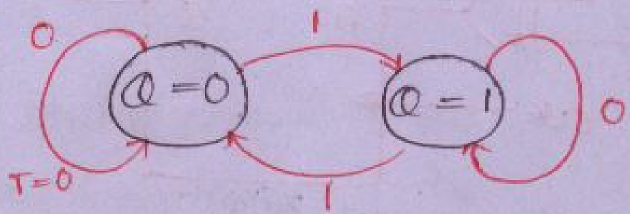
P.S { Q → states



(iii). T- f/f

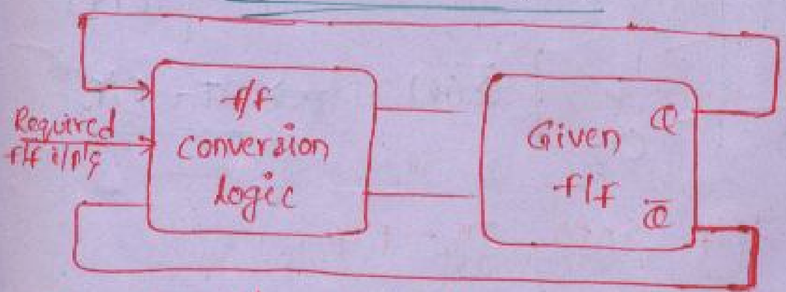
• T → present i/p

Q → P.S.



T	$Q(t+1)$
0	$Q(t)$
1	$\overline{Q}(t)$

CONVERSION OF f/f's :



Q. Convert SR-f/f into T-f/f.

SR f/f → T-f/f

exci. table | char. table

T	$Q(t)$	$Q(t+1)$	S	R
0	0	0	0	x
0	1	1	x	0
1	0	1	1	0
1	1	0		

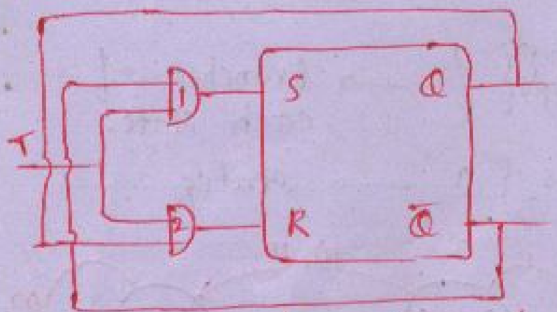
T \ Q	0	1
0	0	X
1	1	0

$S = T\bar{Q}$

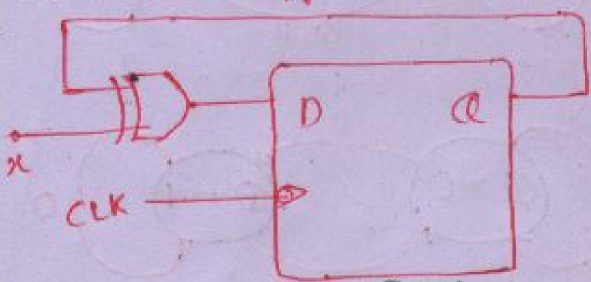
T \ Q	0	1
0	X	0
1	0	1

$R = TQ$

→ T-f/f



Q. Identify the following f/f.



x	Q(t)	$x \oplus Q$ D	Q(t+1)
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

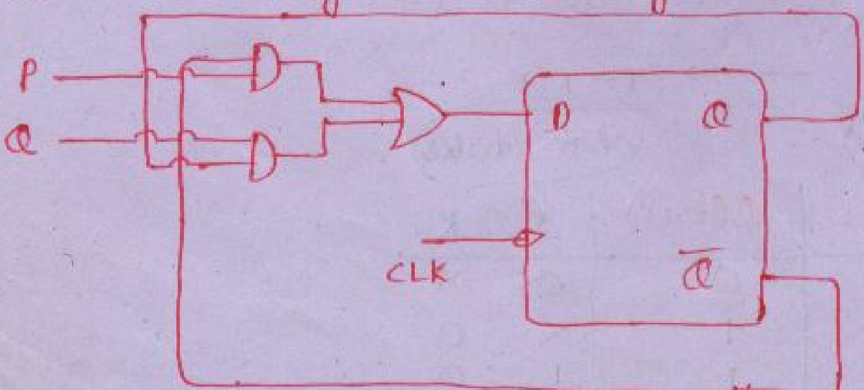
First two rows: Q(t); last two rows: $\overline{Q(t)}$

x	Q(t+1)
0	Q(t)
1	$\overline{Q(t)}$

$\Rightarrow$ T-f/f

Q. Convert D-f/f into JK-f/f.

Q2. Identify the following f/f.



Q. In which of the following counters lockout doesn't occur.

(1). Mod – 13 counter (2). Mod – 30 counter

(3). Mod – 32 " (4). Mod – 36 "

$2^4 - 13 = 3$ unused states

$2^5 - 32 = 0$ unused states

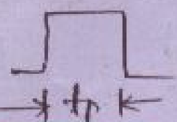
$2^5 - 30 = 2$ unused states

$2^6 - 36 = 28$ unused states

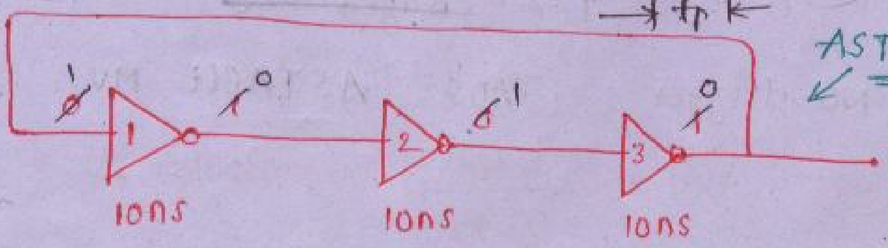
* MON. 08/12/08 *

MULTI VIBRATORS USING LOGIC GATES:

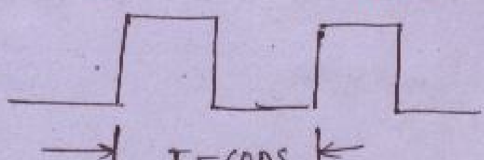
1. ASTABLE MV :
   - → 2 quasi stable states
   - → Square wave Generator.
2. BISTABLE MV :
   - → 2 stable states
   - → 1-Bit memory element
3. MONOSTABLE MV : [one shot]
   - → 1 quasi & 1 stable
   - → pulse generator



ASTABLE MV



30ns

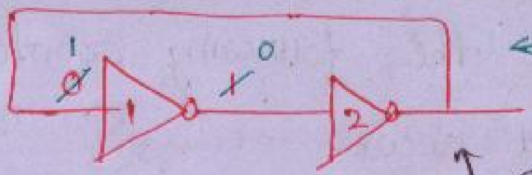


{ Odd no. of inverters }

30ns

T = 2 × (sum of propogation delays of inverters)

(1).

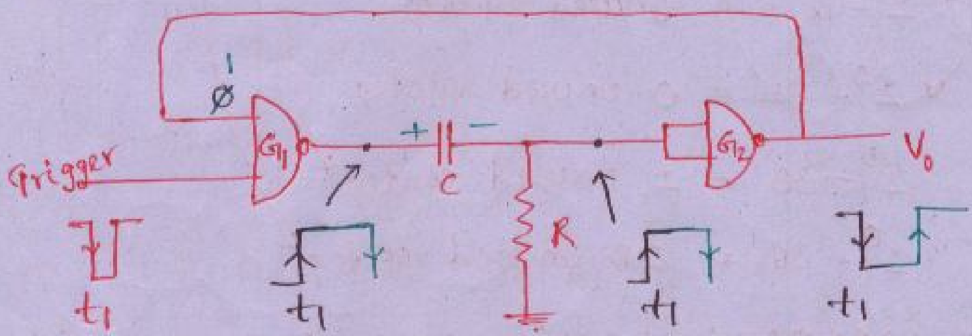


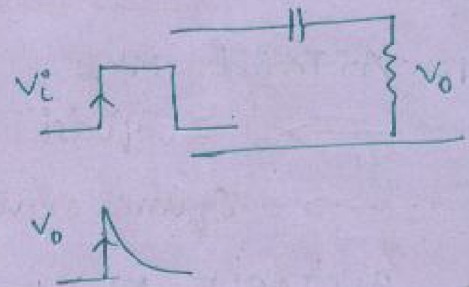
← BISTABLE MV

↑ Even no. of inverters

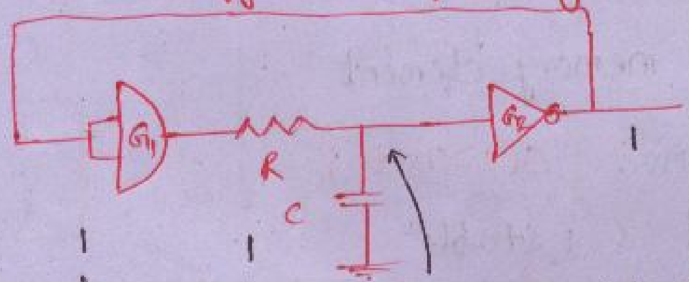
(2). flip flop

MONO STABLE :





Q. Identify the following MV's.

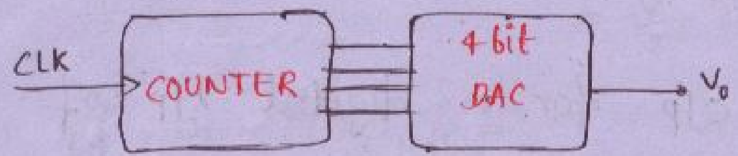


⟶ No trigger ; Ans : ASTABLE MV.

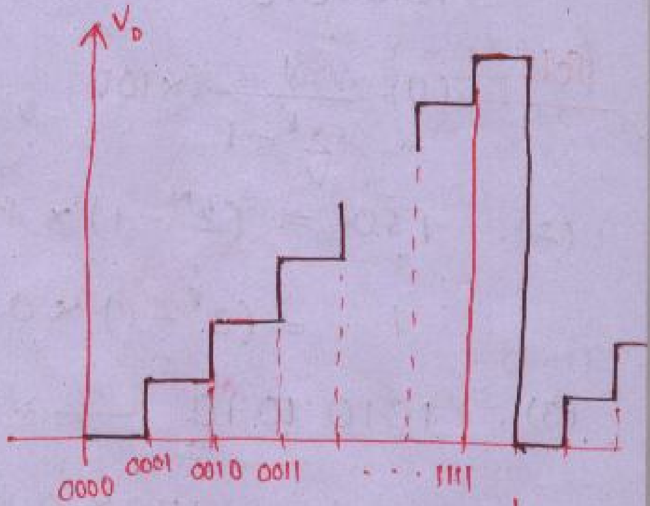
DATA CONVERTERS :

(1). DAC [Digital to Analog]

(2). ADC [Analog to Digital].



CLK	count	V_0
0	0000	0V
1	0001	1V
2	0010	2V
⋮	⋮	⋮
15	1111	15V
16	0000	0V.



no. of steps = 15

FSO (full scale o/p) = 15V

Resolution = step size (V)

It is the smallest possible change at the o/p of DAC for any change in i/p.

'N' bit DAC $\longrightarrow (2^N - 1)$

FSO

$= $ no. of steps $\times$ step size

$= (2^N - 1) \times$ step size.

$\%$ Resolution $= \dfrac{\text{step size}}{\text{FSO}} \times 100$

$= \dfrac{1}{2^N - 1} \times 100$

Q. The o/p of a 8-bit DAC is 0.15 V when the i/p is 00000001.

Determine (1). % resolution

(2). FSO

(3). DAC o/p for a digital i/p of 10101010.

Sol:- (1). $\frac{1}{2^8-1} \times 100$

(2). $FSO = (2^N - 1) \times \text{step size}$

$= (2^8 - 1) \times 0.15\,V$

(3). $10101010_2 \longrightarrow 170_{10}$

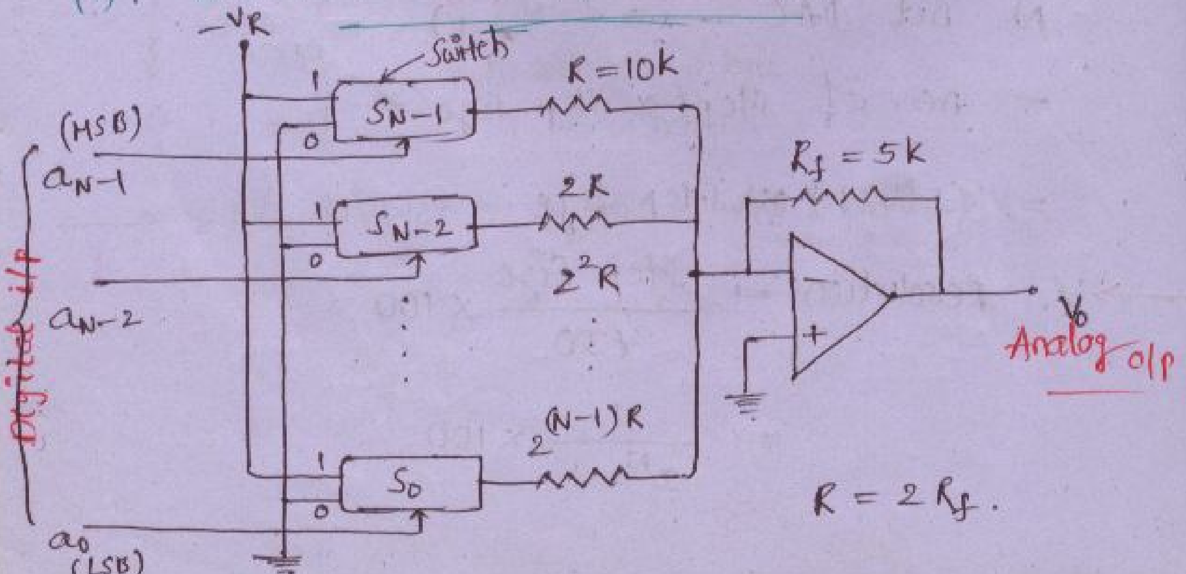
$\therefore$ o/p $= 170 \times 0.15\,V$.

Resolution — voltg (should be less)

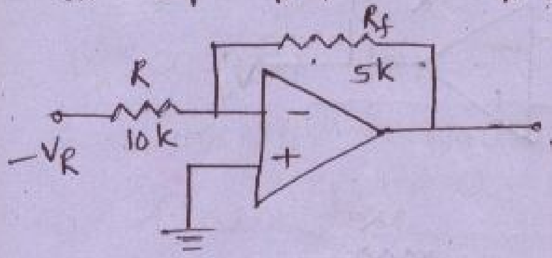
~~0.1V~~
0.5V
1V

8 bit DAC
16 "
~~32~~ "

(1). BINARY WEIGHTED DAC :



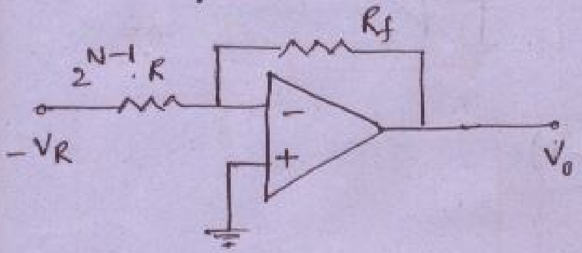
(1). If $a_{N-1} = 1;\ a_{N-2} = \cdots = a_1 = a_0 = 0$



$$= (-V_R)\left(\frac{-5k}{10k}\right)$$

$$= \frac{V}{2}$$

(2). If $a_0 = 1,\ a_{N-1} = \cdots = a_1 = 0.$



$$V_0 = (-V_R)\left(\frac{-5k}{2^{N-1}\cdot 10k}\right)$$

$$= \frac{V_R}{2^N}$$

Resolution

$$V_0 = \left(a_{N-1}2^{-1} + a_{N-2}\cdot 2^{-2} + \cdots + a_1 2^{-(N-1)} + a_0\cdot 2^{-N}\right)V_R.$$

Eg: for a 3 bit DAC.

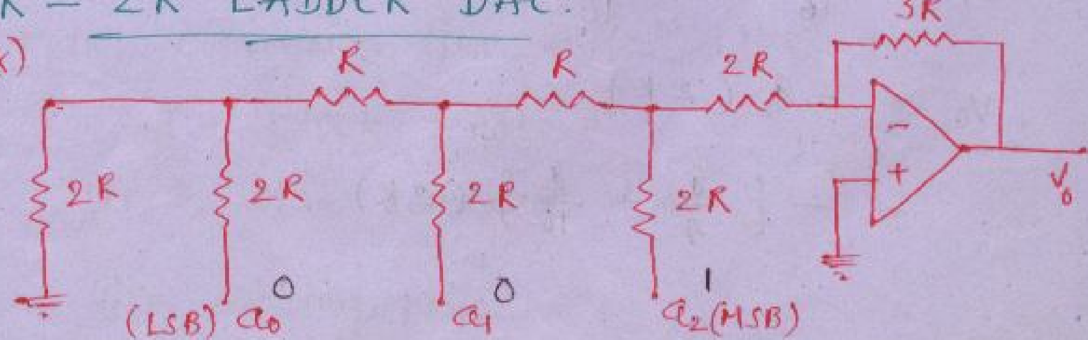
$$V_0 = \left(a_2 2^{-1} + a_1 2^{-2} + a_0 \cdot 2^{-3}\right)V_R$$

$$\text{Resolution} = \frac{V_R}{2^3}.$$

Draw back ⟶ for 32 bit DAC ⟶ $2^{31} \times R$ is required .. and so.

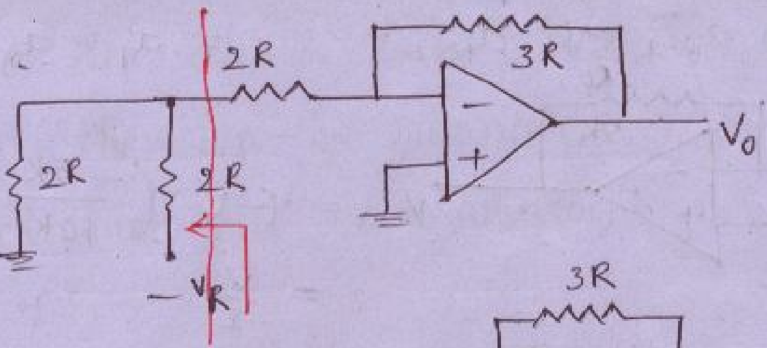
R – 2R LADDER DAC:

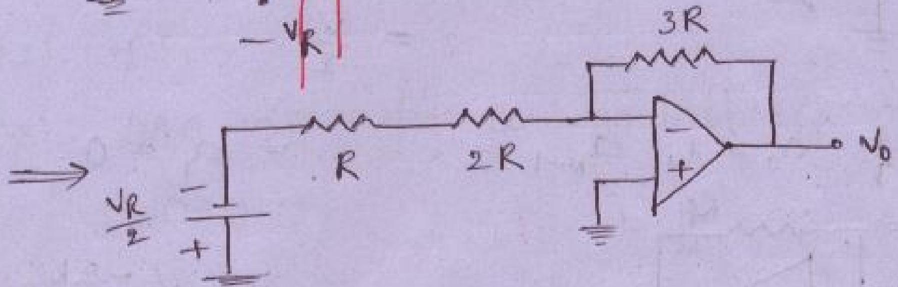
(3-bit)



'1' ⟶ $-V_R$ volts

'0' ⟶ 0 volts.

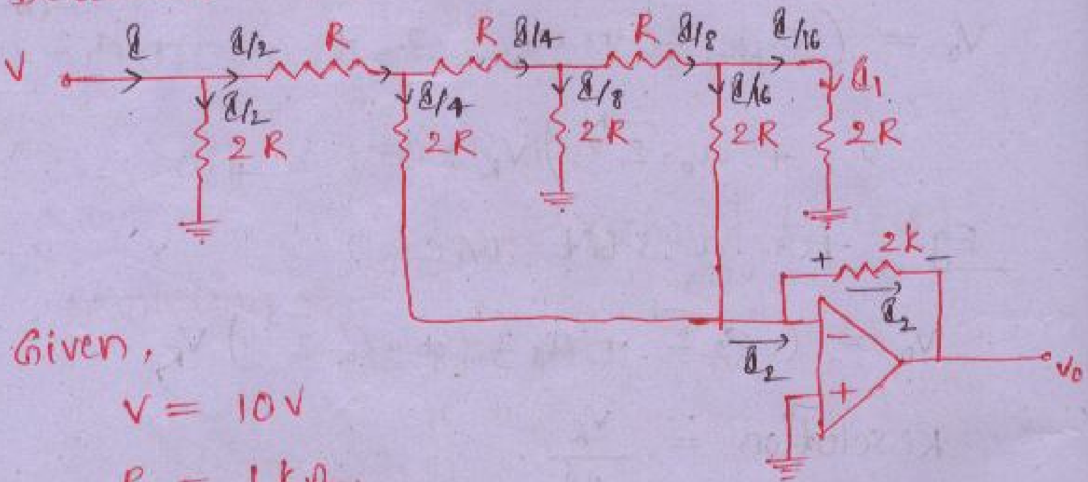




$$V_0 = \left(-\frac{V_R}{2}\right)(-1)$$

$$= \frac{V_R}{2}.$$

Q: Determine I_1 & V_0 in the following circuit.



Given,

$V = 10V$

$R = 1k\Omega$.

$$I = \frac{V}{R} = \frac{10}{1k} = 10\,mA.$$

$$I_1 = \frac{I}{16} = \frac{10\,mA}{16}$$

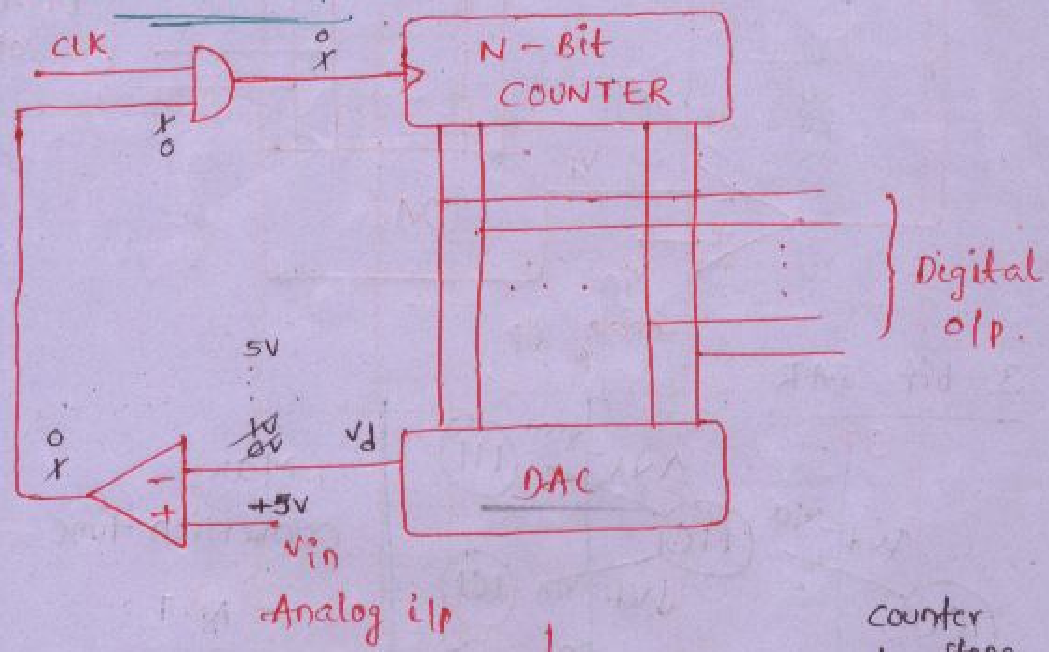
$$V_0 = -I_2(2k)$$

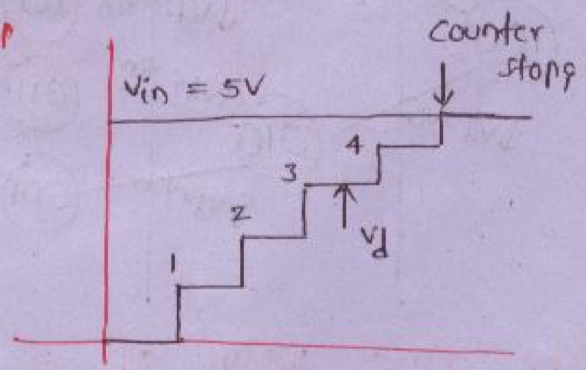
$$= -\left[\frac{I}{4} + \frac{I}{16}\right](2k)$$

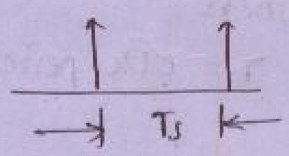
ADC's:

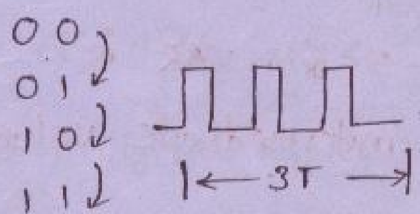
1. counter type
2. Successive approximation type.
3. flash type
4. Dual slope.

COUNTER TYPE:-









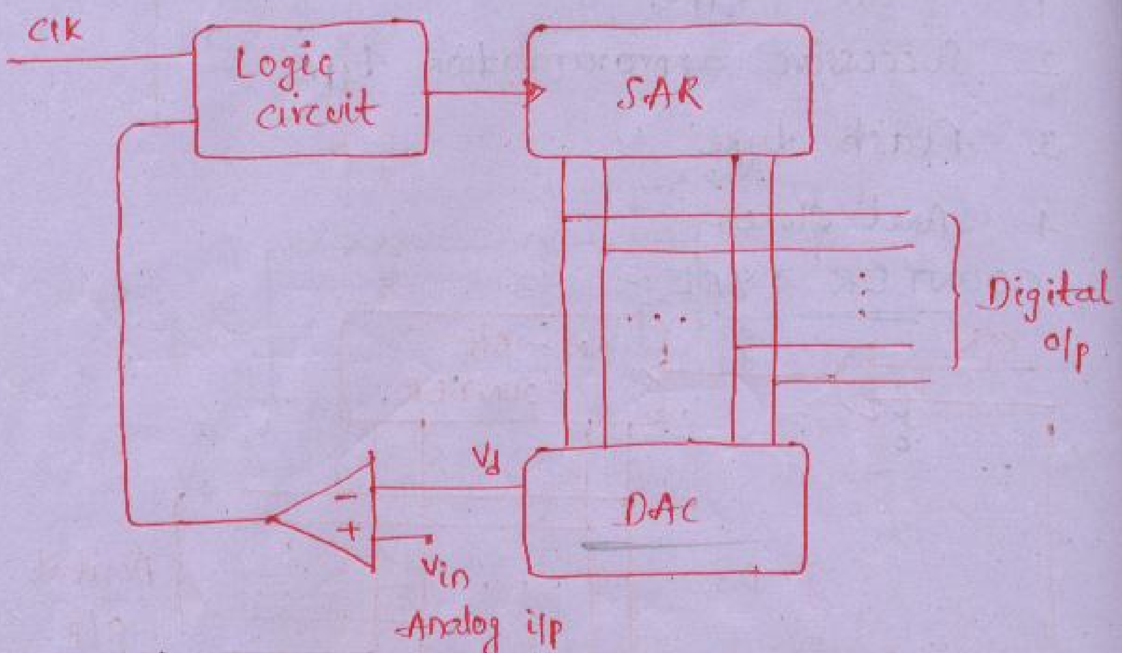
Max. conversion time = $(2^N - 1) \cdot T$;

T – clock period

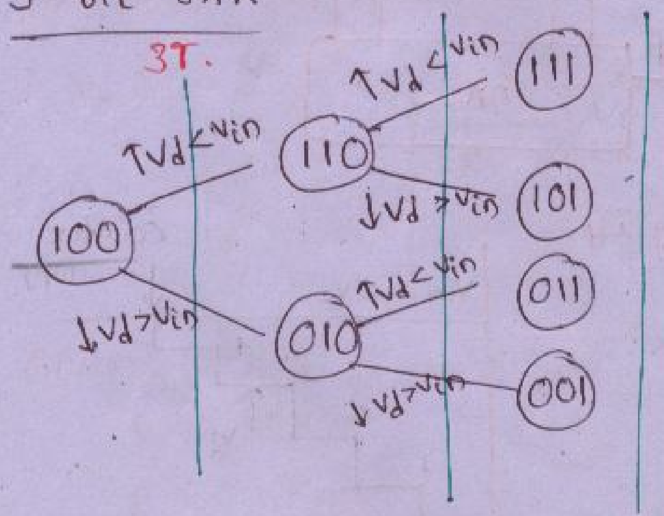
$T_s \geq$ Max. conversion time

T_s = Sampling period

SUCCESSIVE APPROXIMATION ADC :



3-bit SAR



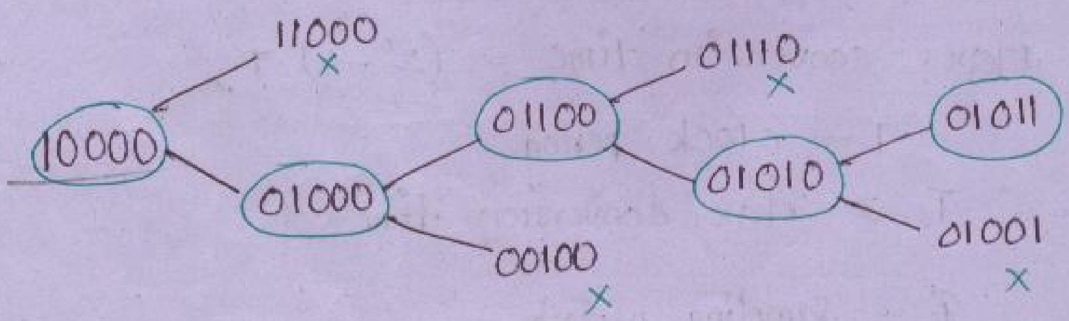
Max. conversion time

$= N \cdot T$

where

$T =$ clk period

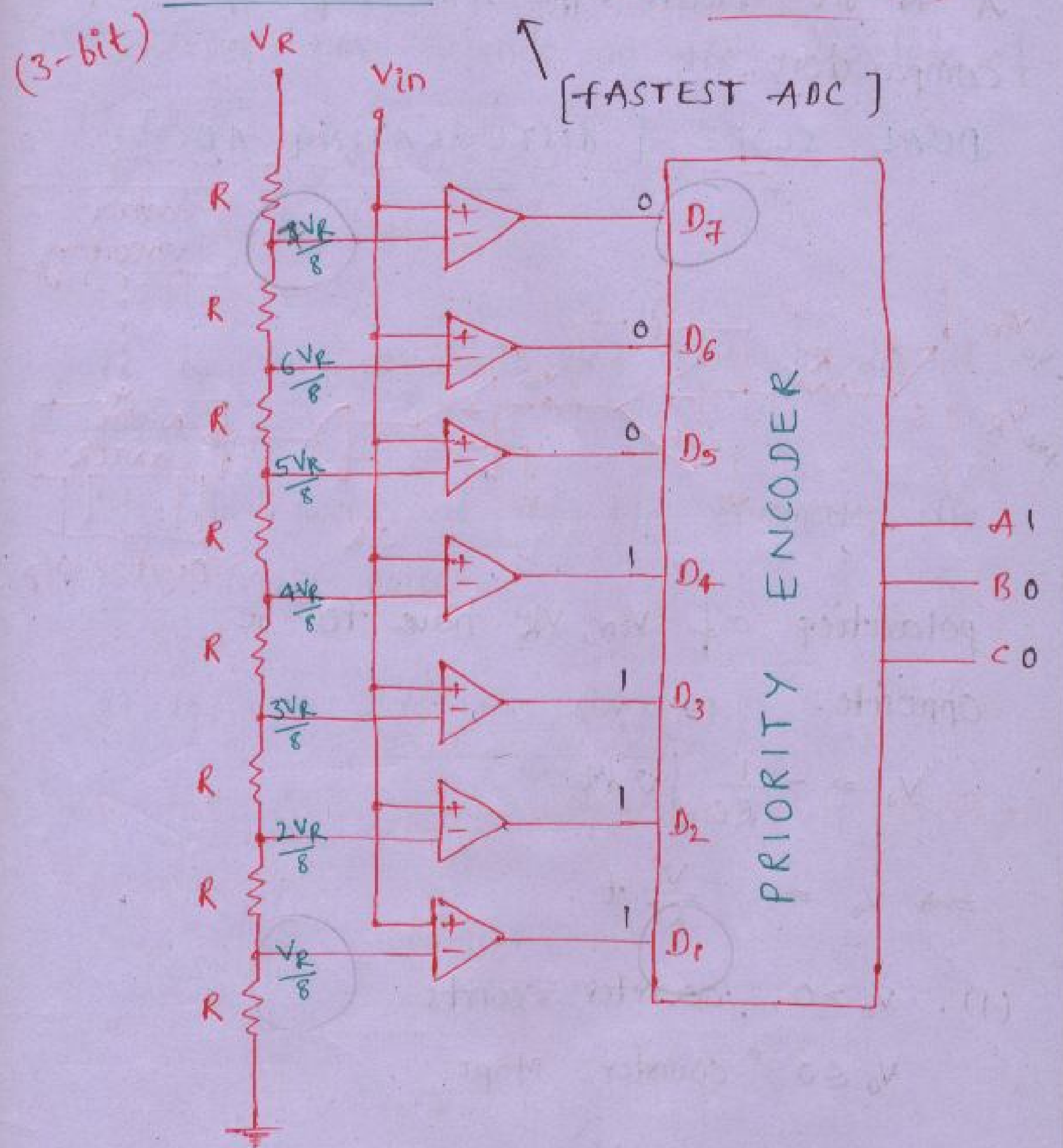
Q. The final value of a 5-bit SAR is 01011. what are its intermediate values ?



FLASH TYPE ADC : [PARALLEL ADC]

[FASTEST ADC]

(3-bit)



let $\frac{4V_R}{8} < V_{in} < \frac{5V_R}{8}$

$\Rightarrow$ Digital o/p = 100

let $V_R = 8$

$4V < V_{in} < 5V$

$\rightarrow$ 100

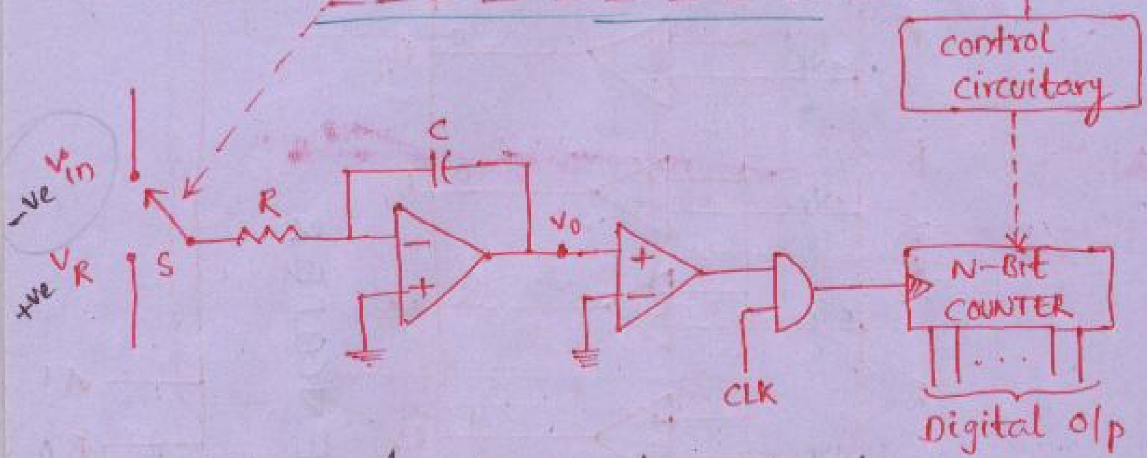
If $\frac{1 \cdot V_R}{8} < V_{in} < \frac{2V_R}{8}$

$\rightarrow$ 001.

Draw back :-

A N-bit flash type ADC requires $2^N - 1$ comparators.

DUAL SLOPE [INTEGRATING ADC] :



polarities of V_{in}, V_R have to be opposite.

$$V_0 = -\frac{1}{RC}\int v\,dt$$

$$\Rightarrow V_0 = -\frac{V}{RC}t.$$

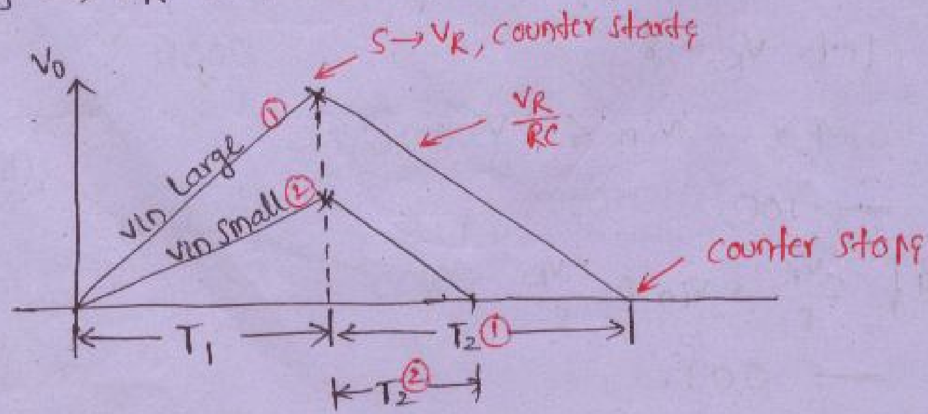
(1). $\dot{V}_0 > 0$, counter counts

$V_0 \leq 0$, counter stops.

(2). control circuitary :

$S \rightarrow V_{in}$ for fixed time 'T_1'

$S \rightarrow V_R$ and counter starts.



→ Max. conversion time $= (2^N - 1)T$.

conversion time depends on the magnitude of i/p.

$$T_2 = \frac{|V_{in}| T_1}{|V_R|}$$

Advantages :

1. It is very accurate and used in digital voltmeters.
2. The integrator at the i/p eliminates the power supply noise.

Draw back :

It is very slow in conversion.

Q. 8-bit ADC, i/p voltage range is -10 to +10

Resolution = ?

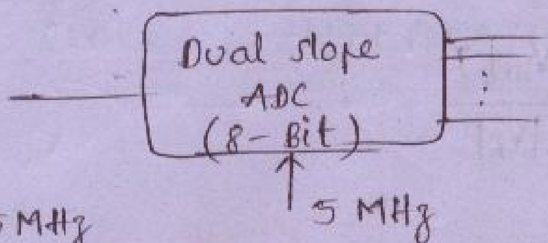
$$\text{Resolution} = \frac{+10-(-10)}{2^8}$$

$$= \frac{20}{256}.$$

Q. To convert $V_{in} = 5V$ into digital, a SAR ADC takes 10 s and dual slope takes 10 ns. Then for $V_{in} = 2.5V$, what is time required ?

$V_{in} = 2.5V$
- SAR ADC ⟶ 10 S
- Dual ADC ⟶ 5 S.

Q. what is sampling rate of 8 bit dual slope ADC if its clk freq. is 5 MHz.



$f = 5\,MHz$

$T = \frac{1}{f} = 0.2\,\mu sec$

$T_s \geq$ max. conversion time

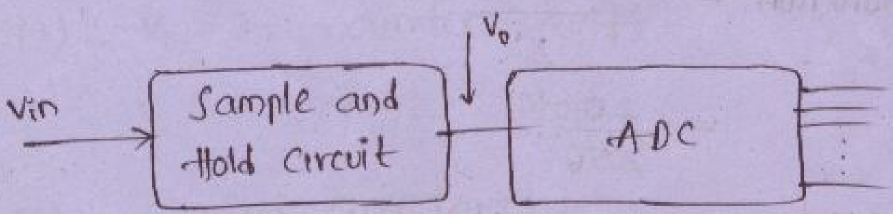
ie $T_s \geq (2^8 - 1)\,T$

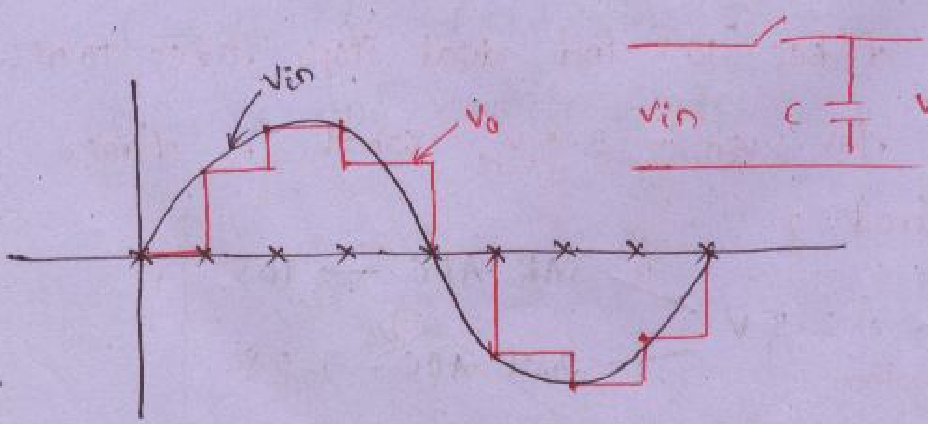
$T_s \geq (255 * 0.2\,\mu sec)$

$T_s = 51\,\mu sec$

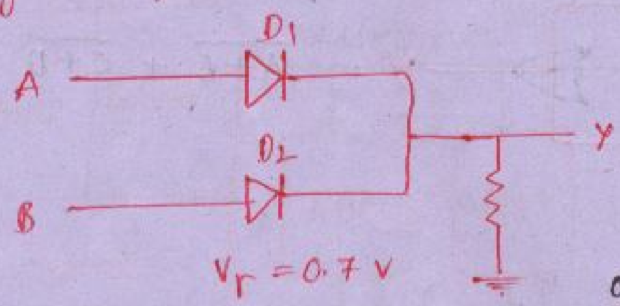
Sampling rate $f_s = \frac{1}{T_s}$

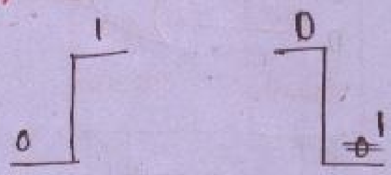
$= \frac{1}{51\mu}$ samples/sec.





Q. Identify the following logic gate in –ve logic –?





A	B	Y
0	0	0
0	+5	$4.3V \simeq 5V$
+5	0	$4.3V \simeq 5V$
+5	+5	$4.3V \simeq 5V$

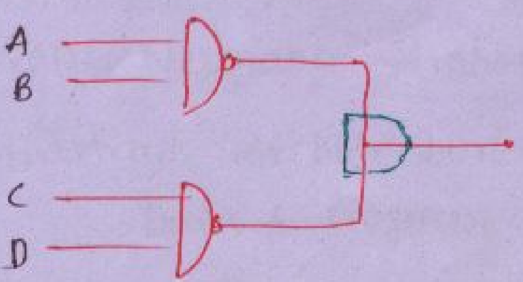
A	B	Y
1	1	1
1	0	0
0	1	0
0	0	0

← AND gate

* The OR gate in +ve logic is equal to AND gate in –ve logic.

+ve logic	–ve logic
NAND	NOR
NOR	NAND
EX–OR	EX–NOR
EX–NOR	EX–OR

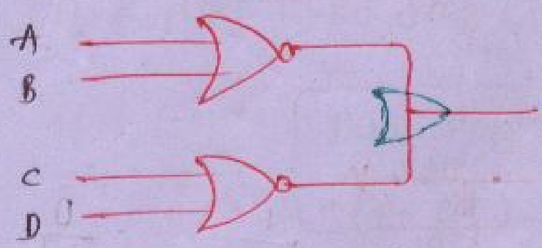
WIRED – AND LOGIC :–



$Y = \overline{AB} \cdot \overline{CD}$

ie $\overline{AB + CD}$

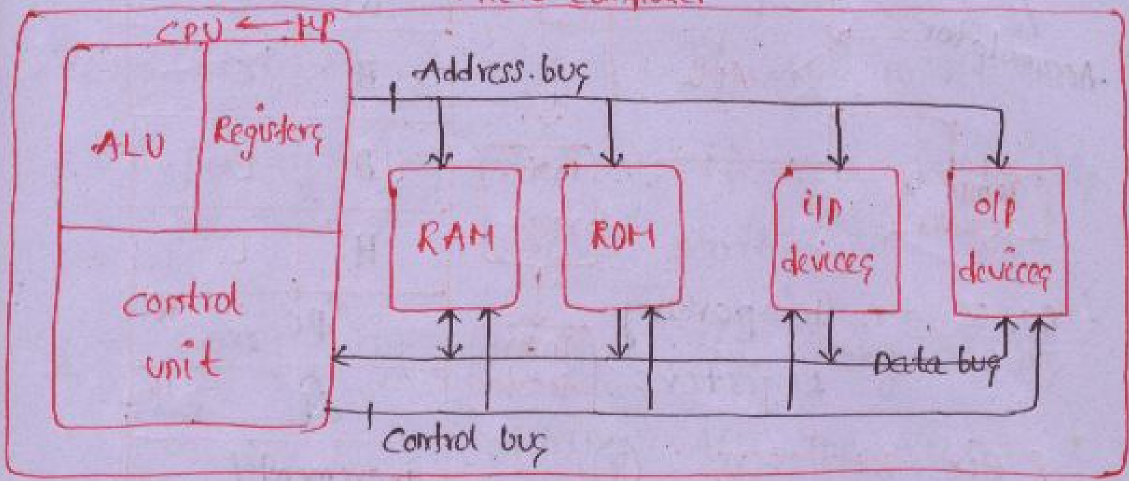
WIRED-OR LOGIC:



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

MICRO PROCESSORS

Micro computer.



8085 µP :-

(1). 16 Adr. lines ⟶ A_0 to A_{15} < $A_8 - A_{15}$, $A_0 - A_7$

Memory capacity $= 2^{16}$

$= 2^6 \cdot 2^{10}$

$= 64 \cdot 1\text{kB}$

$= 64\text{ kB}.$

(2). 8 Data lines ⟶ D_0 to D_7.

(3). freq of µP (f). = ~~[illegible]~~ 3.072 MHz.

(4). Clock freq 'f',

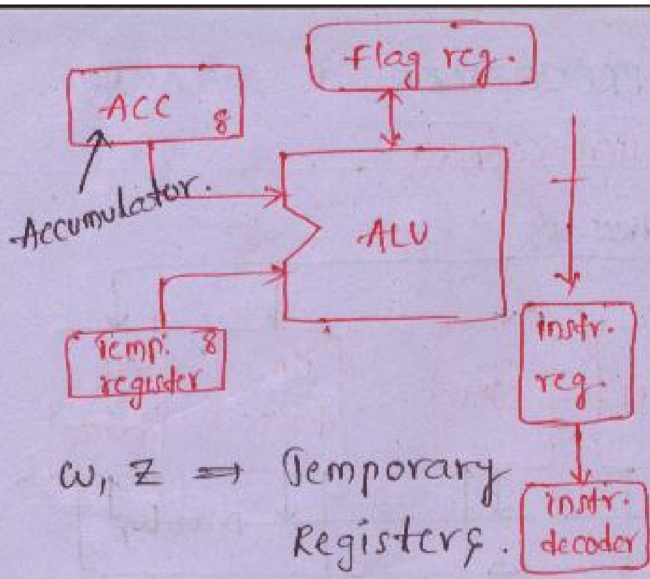
Clock period $T = \frac{1}{f} = 320$ ns.

'NMOS' Tech :

Von Neumann' Architecture ⟶ Data & program stored in the same

Harvard' Architecture ⟶

Data & program are stored seperately



$W, Z \Rightarrow$ Temporary Registers.

REGISTER ARRAY

8 Bit Registers : 7

B
C
D
E
H
L
ACC

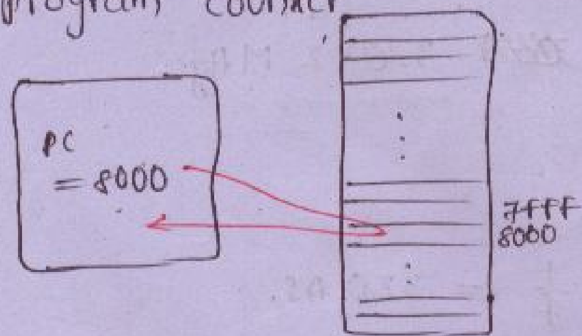
16 Bit Registers : 3

PSW_{16} = program status word
$= ACC_8 + \text{flag reg}_8$

BC
DE
HL ⟶ Memory pointer

PC :

program counter



It indicates memory location from where μp has to fetch its next instr.

FLAG REGISTER :

S	Z	X	AC	X	P	X	CY

S – Sign flag ⇒ S = 1, if MSB of ALU result = 1.

Z – Zero flag ⇒ Z = 1, if ALU result = 0.

P – Parity flag ⇒ P = 1, if ALU result has even parity.

Cy – Carry flag ⇒ Cy = 1, if carry occurs during ALU operations.

AC – Auxilary Carry flag ⇒ AC = 1, if carry occurs from D_3 to D_4 bit.

↳ Can't accessed by the programmer.

⟶ Used in BCD arithmetic operations.

```
   1 1 0 1 1 1 0 1
   1 1 1 1 1 1 0 0
 -----------------
 1 1 1 0 1 1 0 0 1
 -----------------
```

S = 1, P = 0, Z = 0, Cy = 1, AC = 1.

Over flow flag ⟶ Signed Addition

```
+ 0 1 1 (+3) (or)     1 1 0 (−2)          1 0 1
+ 0 1 0 (+2)          1 0 1 (−3)          ↓ 2's
 ------               ------             ------
  1 0 1               0 1 1 ← +ve number  − 0 1 1 ← −3
    ↑ −ve number.
```

⟶ So over flow flag will set in this cases.

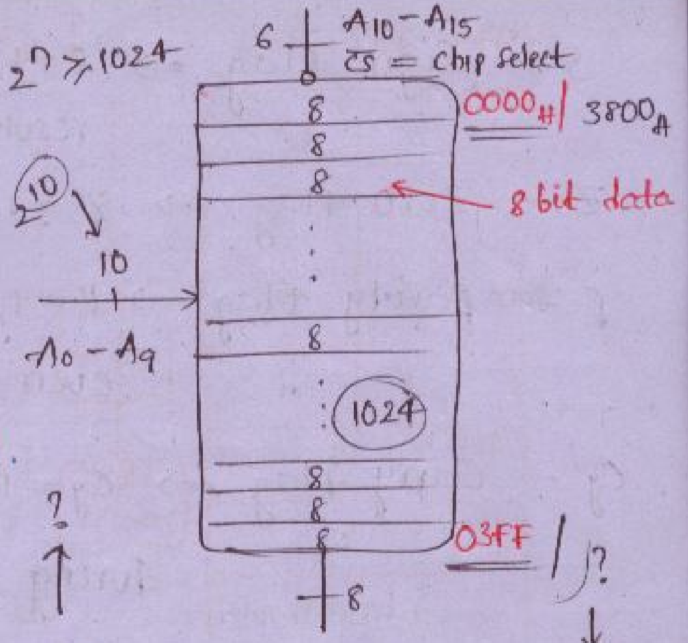
MEMORY IC's:

1 kB Memory

$= 1024 \times 8$

$2^n \geq 1024$

2^{10}



03FF ← 16 bit Address

$= 0000\ 0011\ 1111\ 1111$.

$$\begin{array}{r} 8fff \\ -03ff \\ \hline 8c00 \end{array}$$

? ↑ 8fff

$$\begin{array}{r} 03ff \\ 3800 \\ \hline 3BFF \end{array}$$

4 kB Memory

4 kB

$= 2^2 \cdot 2^{10}$

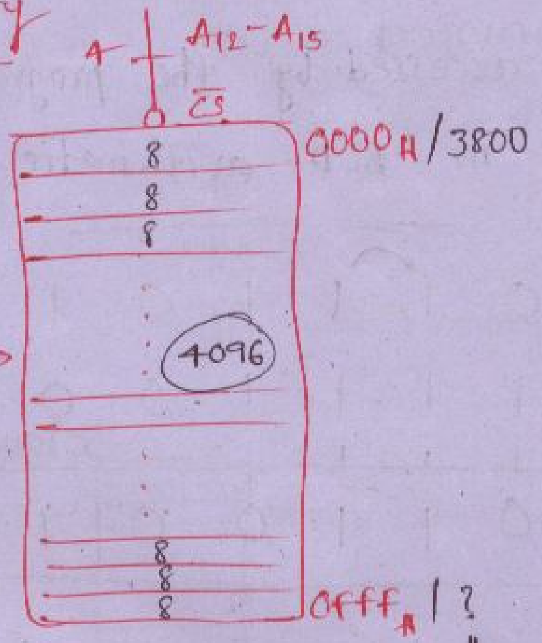
$= 2^{12}$

$= 4 \times 1024 \times 8$

$(2^n \geq 4096)$

$n = 12$

$= 4096 \times 8$



$$\begin{array}{rr} f = & 15 \\ 8 & 8 \\ \hline 17 & 23 \end{array}$$

$$\begin{array}{r} 0fff \\ 3800 \\ \hline 47ff \end{array}$$

Q. For a 32 kB memory the ending location address is "Afff". What is its starting address. ?

Ans: 3000_H

EPROM ... RAM

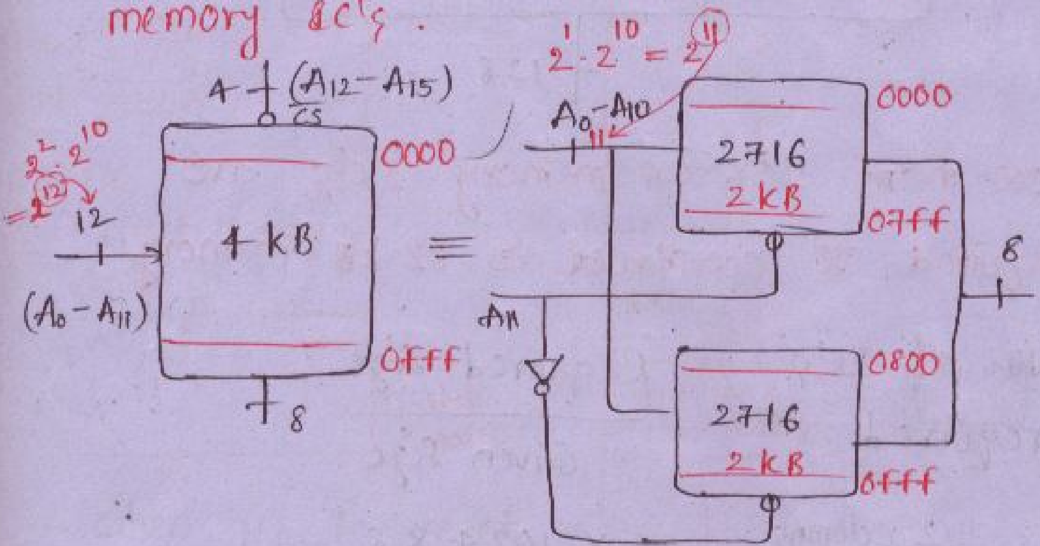
$\boxed{27}16 = 2$ kB ← $\boxed{61}16$

2732 = 4 kB ← 6132

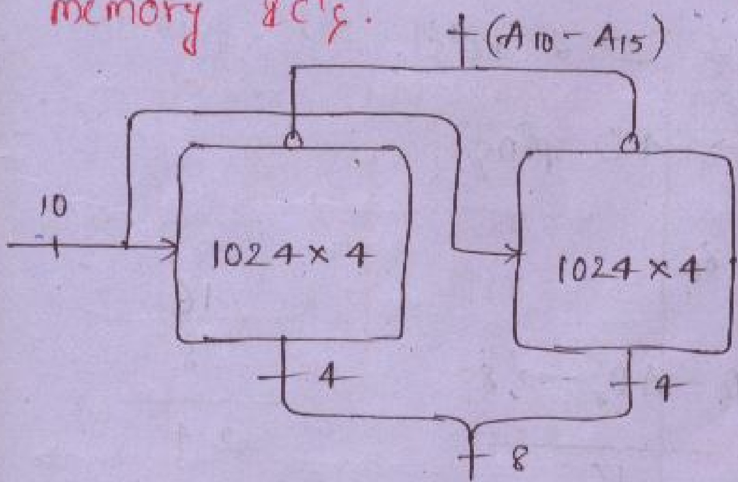
2764 = 8 kB ← 6164

27128 = 16 kB ← 61128

Q. Construct a 4 kB memory using 2716 memory IC's.



Q. Construct a 1kB memory using 1024×4 memory IC's.



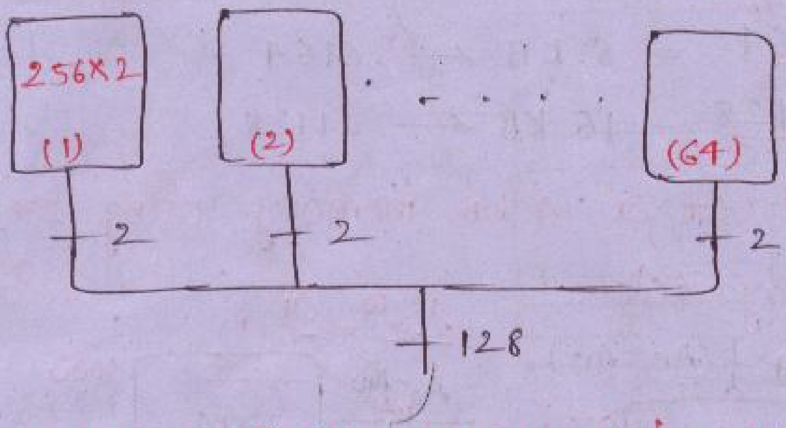
$\frac{1\text{kB}}{1024 \times 8}$

Q. The add. lines of 64 memory IC having capacity of 256×2 are connected together. What is the size of resulting memory.

64 Memory IC's.

256 × 2

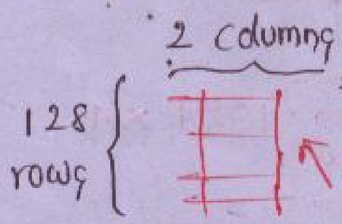
$256 \times (64 \times 2)$
$= 256 \times 128$



Q. How many 256 ×4 memory IC's are required to construct a 32 kB memory.

$$\text{No. of IC's required} = \frac{\text{Required size}}{\text{Given size}}$$

$$= \frac{32 \times 1024 \times 8}{256 \times 4}$$



$= 256$ IC's.

8085 μP → 40 pins

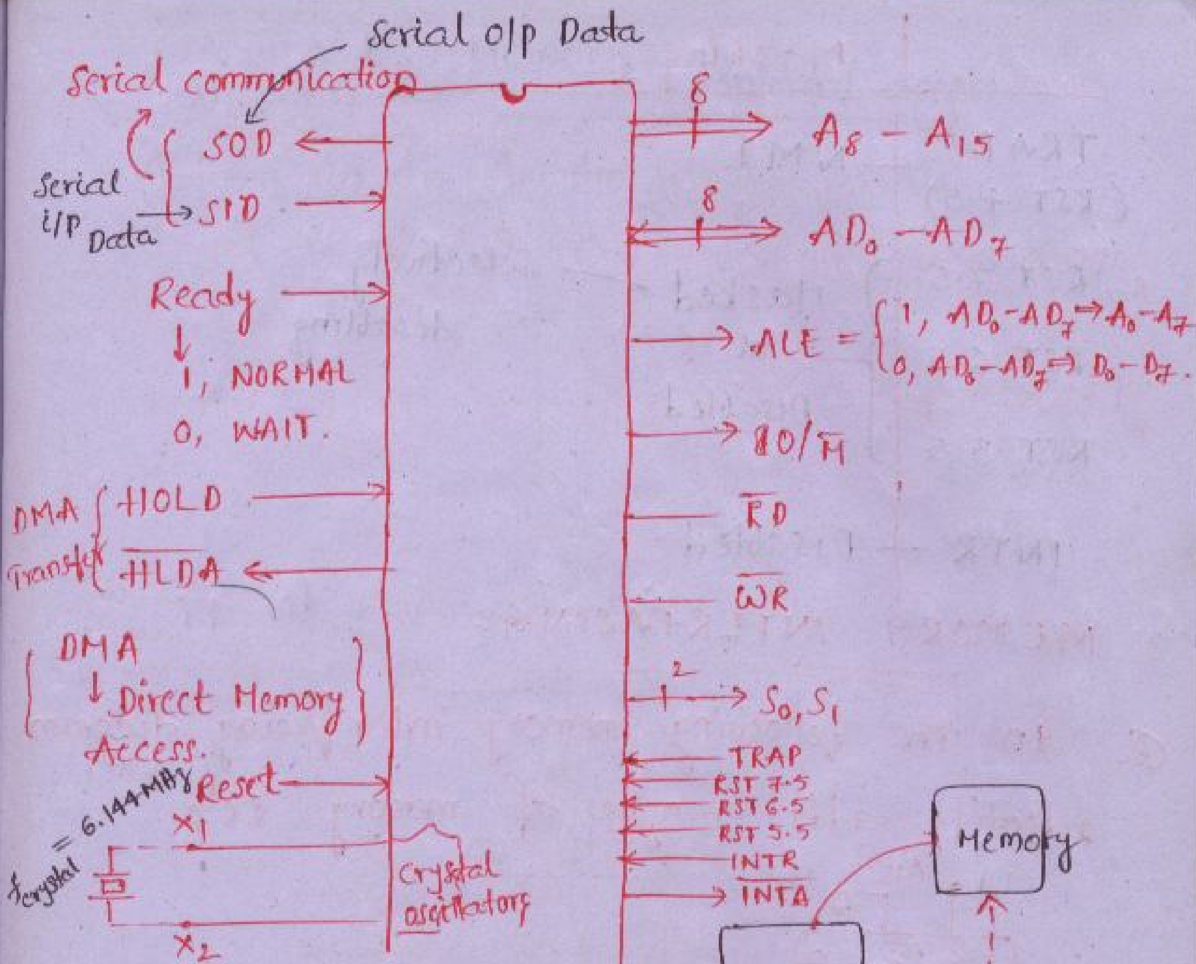
$A_8 - A_{15} \rightarrow 8$

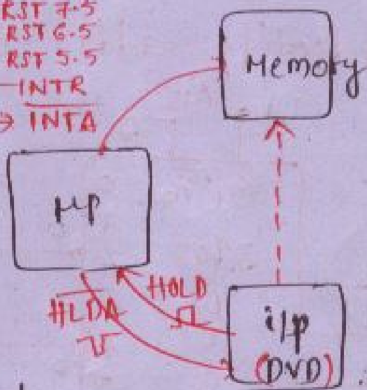
$A_0 - A_7$, $D_0 - D_7$ } $AD_0 - AD_7 \rightarrow 8$

16

16
8
24

* Ready pin used to interface μp with slow speed peripherals.

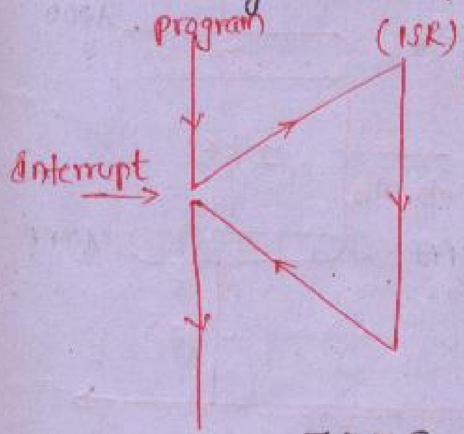




when µp is reset,

all the reg.s get cleared

including the program counter.



(RST 4.5)	Priority	Vector address
TRAP	Highest	0024_H
RST 7.5		003C
RST 6.5		0034
RST 5.5		002C
INTR	lowest	(Non vector Interrupt)

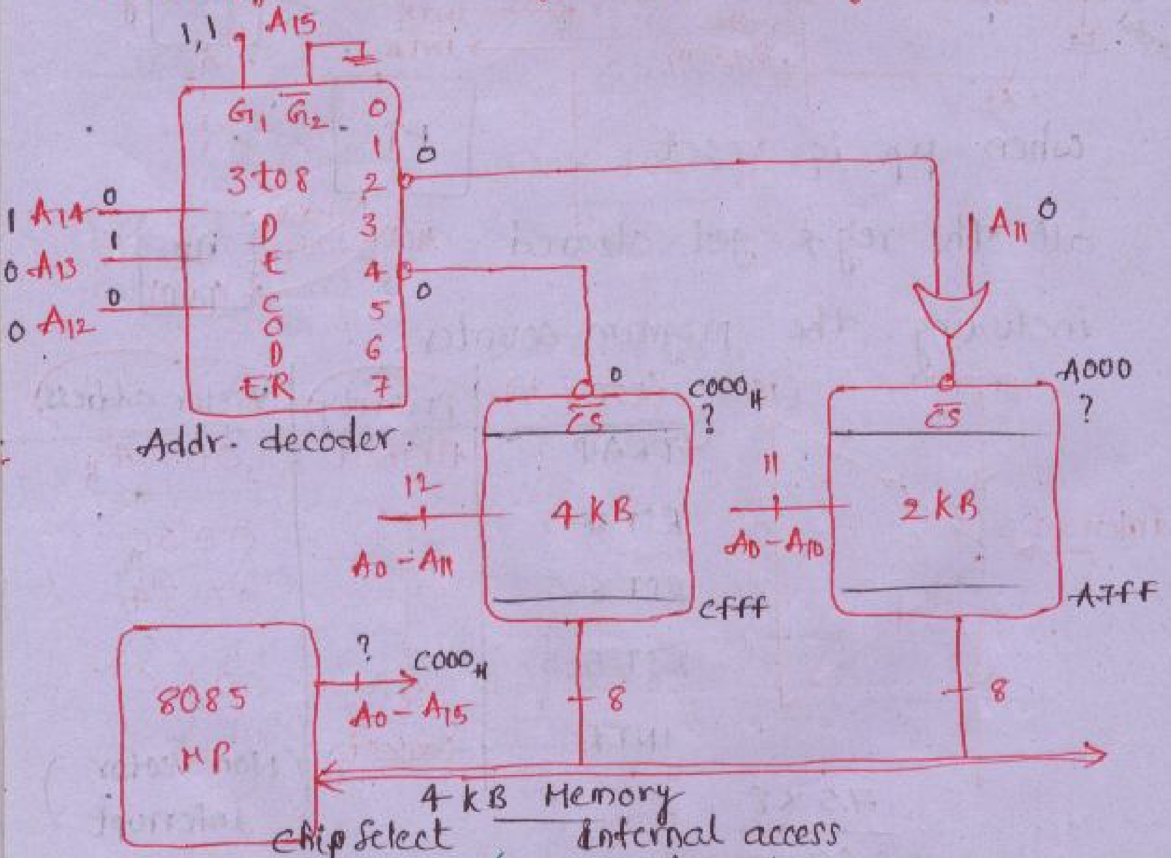
$$\frac{7.5 \times 8}{60_{10}}$$
$$\downarrow$$
$$3C_H.$$

Masking

TRAP (RST 4.5)	NMI
RST 7.5, RST 6.5, RST 5.5	Masked or Disabled ← selectively disabling
INTR	Disabled

MEMORY INTERFACING:

Q. In the following memory interfacing diagram identify addr. ranges of memory IC's.

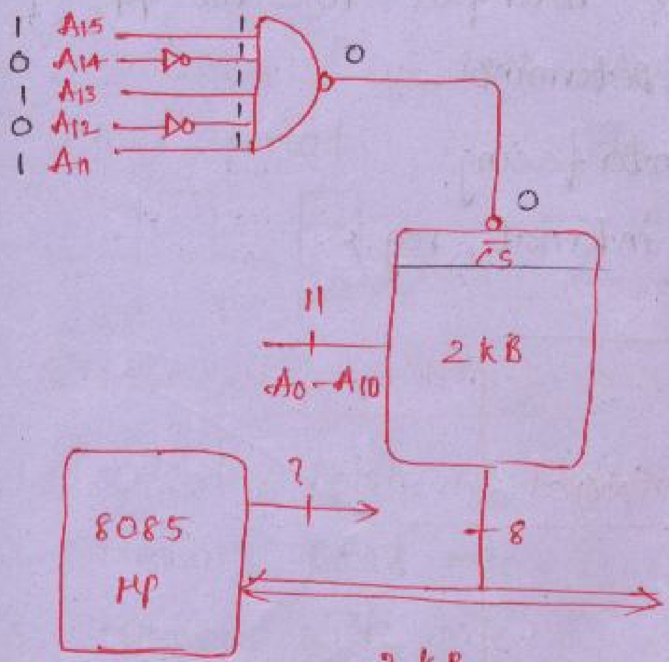


Chip select				4 KB Memory internal access			
A_{15}	A_{14}	A_{13}	A_{12}	A_{11} . . . . . A_0		
1	1	0	0	0 0 0	=	$C000_H$
1	1	0	0	1111 1	=	$CFFF_H$.

$$\underbrace{A_{15}\ A_{14}\ A_{13}\ A_{12}\ A_{11}}_{\text{chip select}}\ \overbrace{\ }^{\text{2 KB Memory}}\ \underbrace{A_{10}\ \ldots\ldots\ A_0}_{\text{Internal access}}$$

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	. . .	A_0	
1	0	1	0	0	0	. . .	0	$= A000_H$
1	0	1	0	0	1	1 . . .	1	$= A7FF_H$

Q.



$$\underbrace{A_{15}\ A_{14}\ A_{13}\ A_{12}\ A_{11}}_{CS}\ \underline{\underline{2\ KB}}\ \underbrace{A_{10}\ A_9\ \ldots\ldots\ A_0}_{\text{Internal access}}$$

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	. . .	A_0	
1	0	1	0	1	0	0	. . .	0	$= A800_H$
1	0	1	0	1	1	1	. . .	1	$= AFFF_H$

I/O INTERFACING:

1. Memory mapped I/O → I/O devices are considered as memory I/C.
2. I/O mapped I/O. ↘ I/O & Memory are considered seperately.

	Memory mapped I/O: Memory	Memory mapped I/O: I/O	I/O mapped I/O: memory	I/O mapped I/O: I/O
1). No. of addr. lines	16	16	16	8 (A-…)
2). control signals	$\overline{MEMR}$ $\overline{MEMW}$	$\overline{MEMR}$ $\overline{MEMW}$	$\overline{MEMR}$ $\overline{MEMW}$	$\overline{IOR}$ $\overline{IOW}$
3). No. of peripherals	60 kB → 4 kB 50 kB → 14 kB 64 kB → Nil			$2^8 = 256$ I/O devices

control signals :

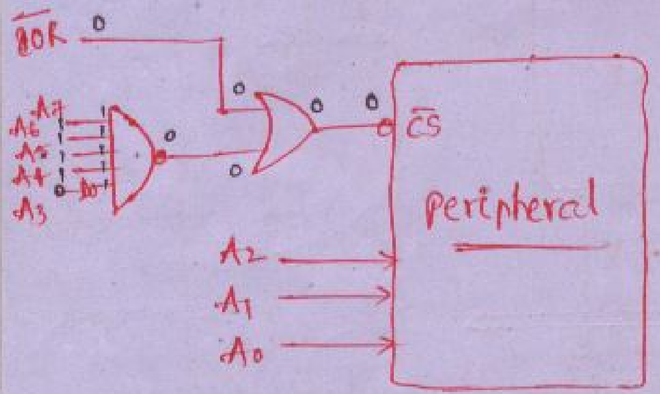
$\overline{MEMR}$ $\overline{IOR}$

$\overline{MEMW}$ $\overline{IOW}$

Q. A peripheral is interface to the μP as shown below. Determine —

(1). Mode of interfacing

(2). Addr.s of internal reg.s.



Ans: (1). I/O mapped I/O. mode.

(2). peripheral has 8 reg.s., peripheral is i/p device b'coz $\overline{IOR}$

	A7	A6	A5	A4	A3	A2	A1	A0	
R1	1	1	1	1	0	0	0	0	= F0
R2					.				
R3					.				
R4					.				
R5					.				
R6					.				
R7					.				
R8	1	1	1	1	0	1	1	1	= F7

INSTRUCTION CYCLE :

Time required to execute an instr.

Range : 1 machine cycle to 5 m/c.

MACHINE CYCLE :

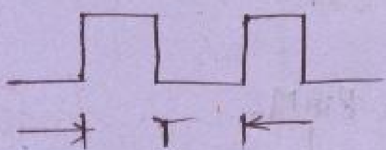
MP — memory   MP — i/o

MP $\xrightarrow{ACK}$ Time required to complete one operation of accessing memory, accessing I/O devices & sending an acknowledgement.

Range : 3 T states to 6 T.

T - STATE :

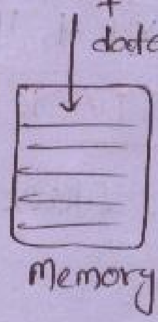
It is sub task performed in one clock period.



T = 320 ns

TYPES OF MACHINE CYCLES:

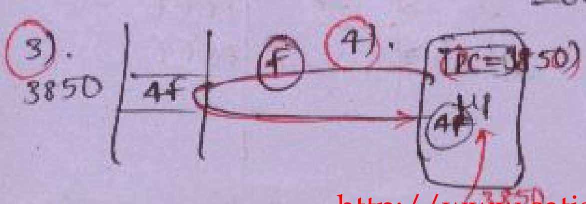
1. Opcode fetch m/c — (F) → 4T
2. Memory Read m/c — (R) → 3T
3. Memory write m/c — (W) → 3T
4. I/O Read m/c — (I) → 3T
5. I/O write m/c — (O) → 3T
6. Hold ACK m/c
7. Interrupt ACK m/c



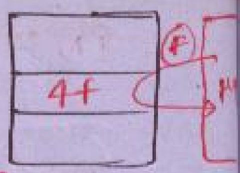
Opcode fetch m/c → 4T = 3T + 1T

decoding (1T), fetching (3T)

(1). MOV C, A → (2). opcode $= 01001111_2 = 4F_H$

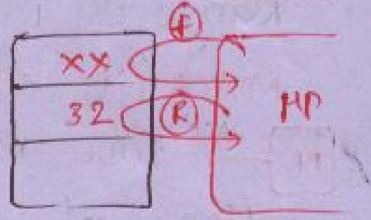
(3). 

1 – Byte Instruction : ⟶ MOV C, A



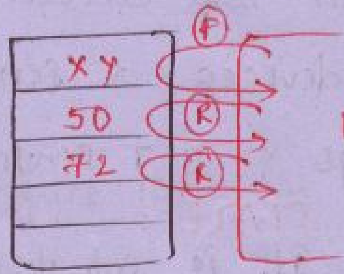
2 – Byte Instruction :

⟶ ~~MOV~~ MVI C, 32
let xx



3 – Byte Instruction :

⟶ LDA 7250
xy



XTHL ⟶ 1B

ANI f2 ⟶ 2B

LDAX B ⟶ 1B

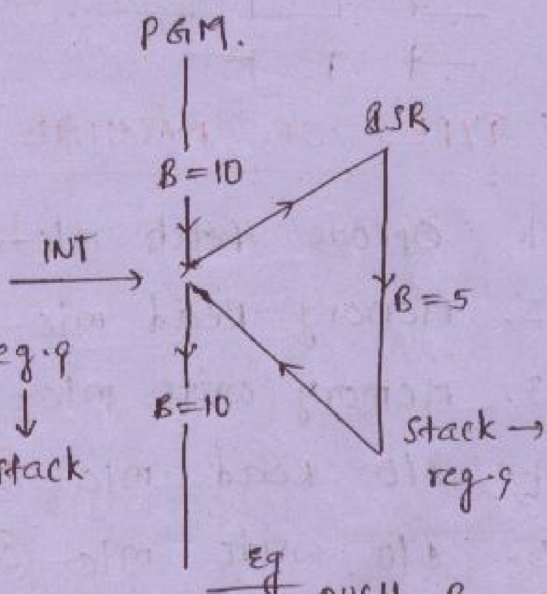
LXI H, 1122 ⟶ 3B

STACK :

SP : Stack pointer

LIFO :

Last in First Out



(1). PUSH RP
↑ reg. pair

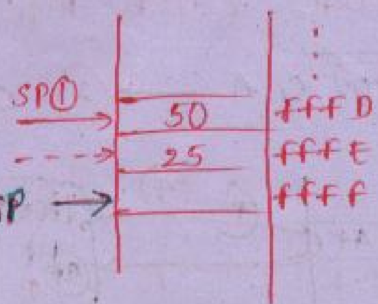
Eg
PUSH B
PUSH D
PUSH H

⤷ decrement sp + push higher reg. (H)
decrement sp + push lower reg. (L)

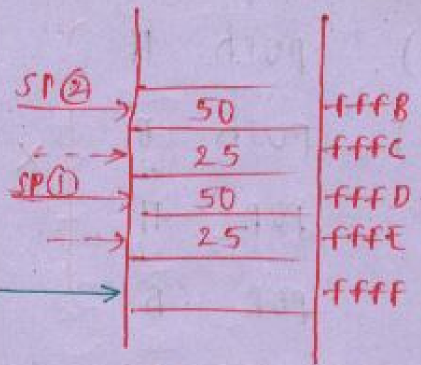
Eg : Let sp = ffff

HL = 2550
H L

1). PUSH H SP
fffD



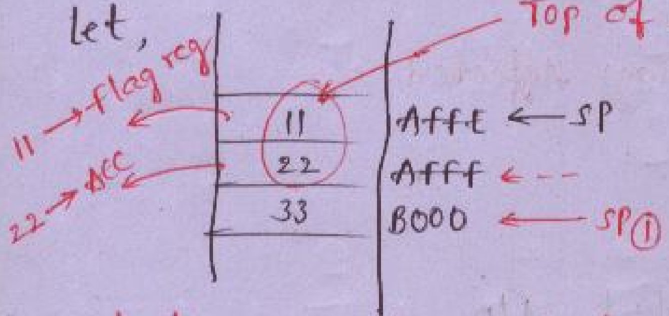
2). PUSH H $\frac{SP}{FFFB}$



POP rp :

Get 1 Byte into lower reg + Increment SP
Get 1 Byte into higher reg + Increment SP

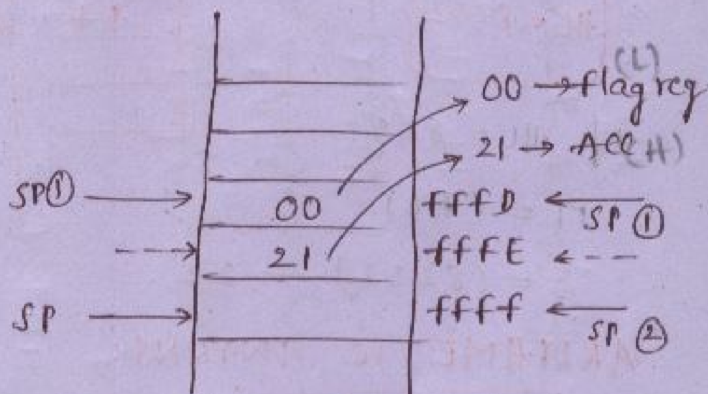
let,



POP PSW

Q. what are the contents of Acc & flag reg after executing following instructions.

(1). SP = FFFF (i). Push H
HL = 2100 (ii). POP PSW



The above program is used to clear the flag register.

1). push H
2) push B
pop H
pop B
×

2). push H
push B
pop B
pop H
✓

INSTRUCTIONS:

1. Data Transfer
2. Arithmetic
3. Logical

Arithmetic and Logical: Flags Affected.

4. Branching
5. Machine related, I/O
6. Additional

µP
$BC = 8250$
$HL = 8252$

Memory data	Addr.
11	8250
22	8251
33	8252
44	8253

$(BC) = (8250) = 11$

$(HL) = (8252) = 33.$

$M = (HL) = 33.$

If $HL = 8252$

$M = (HL)$

$= (8252) = 22.$

ARITHMETIC INTRNS:

M/C → get the instr + operation

F 1 + 0

1 + 1

ADI 47 2 + 0

	Instruction	Operation	Bytes/M/c/T	Types of M/c	flags affected
1).	ADD R (R → B, C, D, E, H, L, A.)	$A + R \rightarrow A$	1/1/4	F	ALL
	ADD M	$A + (HL) \rightarrow A$	1/2/7	F, R	ALL
	ADI 8bit data	$A + (8\text{bit data}) \rightarrow A$	2/2/7	F, R	ALL
2).	SUB R				
	SUB M				
	SUI 8bit data				
3).	ADC R	$Cy + A + R \rightarrow A$	1/1/4		
	ADC M	$Cy + A + (HL) \rightarrow A$	1/2/7		
	ACI 8bit data	$Cy + A + (8\text{bit data}) \rightarrow A$	2/2/7		
4).	SBB R				
	SBB M	$A - (HL) - Cy \rightarrow A$	1/2/7		
	SBI 8bit data				

5). INR R — $R+1 \rightarrow R$ — 1/1/4 — F — ALL except 'cy' flag

DCR R — $R-1 \rightarrow R$ — 1/1/4 — F

INR M — $(HL)+1 \rightarrow (HL)$ — 1/2/10 — F, R, W — ALL except 'cy' flag

DCR M — $(HL)-1 \rightarrow (HL)$ — 1/2/10 — F, R, W

S = opcode fetch m/c (6T)

INX rp — $rp+1 \rightarrow rp$ — 1/1/6 — S

B = Bus Idle m/c (3T) — NO FLAGS

DCX rp — $rp-1 \rightarrow rp$ — 1/1/6 — S

BC = 8250,

INR B	INX B
B = 83	BC = 8251.

6). DAD rp — $HL+rp \rightarrow HL$ — 1/3/10 — F, B, B — only 'cy' flag.

LOGICAL INSTRUCTIONS:

1). ORA R — $A \vee R \rightarrow A$ — ALL but $C_y = 0$.

ORA M — $A \vee (HL) \rightarrow A$

ORI 8bitdata — $A \vee$ 8bitdata $\rightarrow A$

2). ANA R

OR = ∨

AND = ∧

XRA = ⊕

ANA M   A ∧ (HL) → A

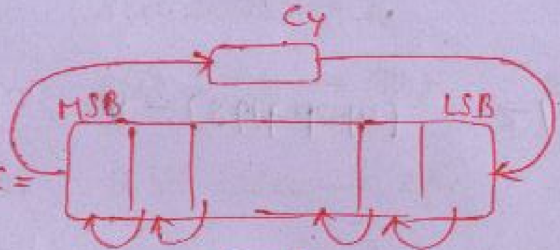
AN8

3). XRA R   A ⊕ R → A

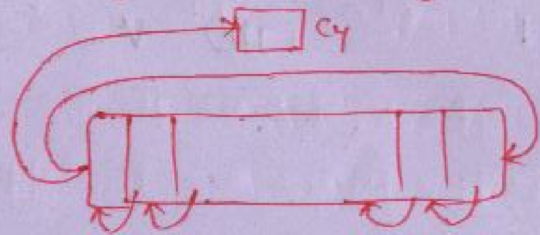
XRA M

XR8 8bitdata

4). RAL
(with Cy) ACC =

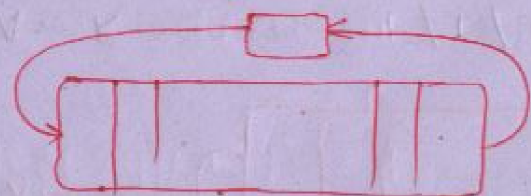


RLC
(without Cy)

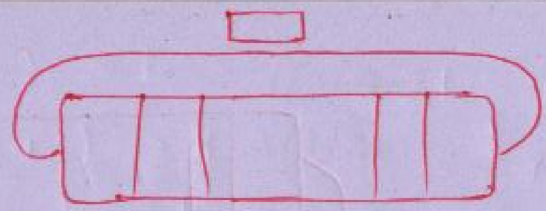


Affects only Cy.

5). RAR
(with Cy) ACC =



RRC (without cy)

ACC. = 

6).	CMP R	$A - R$	1/1/4	F	$A > R$: $cy = 0, Z = 0$ $A < R$: $cy = 1, Z = 0$ $A = R$: $cy = 0, Z = 1$ S, P, AC affected
	CMP M	$A - (HL)$	1/2/7	F, R	– do – exp. $R \rightarrow (HL)$
	CPI 8bitdata	$A - (8\text{bit data})$	2/2/7	F, R	– do –
7).	CMA	$\bar{A} \rightarrow A$	NO flags		
	CMC	$\bar{cy} \rightarrow cy$	only 'cy'		
	STC	$cy = 1$	only 'cy'		

$A = F2$

$\boxed{\text{ANI 0F}}$ Masking

$A = ?$

$A \rightarrow$ 1111 0010
0000 1111
0000 0010

02

DATA TRANSFER INSTR.S :

1). MOV R_d, R_s	$R_s \longrightarrow R_d$		
MOV R, M	$(HL) \rightarrow R$		
MOV M, R	$R \rightarrow (HL)$		
MVI ~~MOV~~ R, 8bitdata	8bit data $\rightarrow$ R		
MVI M, 8bitdata	8bit data $\rightarrow$ (HL)	2/ $\underset{\uparrow}{3}$ / 10 (FR W: 2+1)	F, R, W
2). LXI rp, 16bitdata (Load immediate)	16 bit data $\rightarrow$ rp	3/ $\underset{\uparrow}{3}$ / 10 (FRR: 3+0)	F, R, R
3). LDA 16bit address (Load Accumulator)	(16 bit addr.) $\rightarrow$ A	3/4/13	FRR R
4). STA 16 bit address (Store Accumulator)	A $\rightarrow$ (16 bit addr.)	3/4/13	FRR W
5). LDAX rp	(rp) $\rightarrow$ A	1/2/7	FR
STAX rp	A $\rightarrow$ (rp)	1/2/7	FW

6). LHLD 16 bit addr. (16 bit addr.) $\rightarrow$ L 3 | 5 / 16 fRR RR

(16 bit addr + 1) $\rightarrow$ H

$\downarrow$ 3+2

fRR RR

LHLD 8252 ie (8252) $\rightarrow$ L

(8253) $\rightarrow$ H

SHLD 16 bit addr. L $\rightarrow$ (16 bit addr.) 3 | 5 | 16 fRR WW

H $\rightarrow$ (16 bit addr. + 1)

SHLD 8254 $\updownarrow$, (HL) = 8090

44	8253
90	8254
80	8255