Number System

1. Binary Number system
2. Octal Number system
3. Hexadecimal number system

Use of Hexa-decimal number:

used in assembly language programming. The instruction and data are entered as Hexadecimal numbers because instruction code and data are very lengthy

Conversion

$(.36)_{10} \longrightarrow (.010111)_2$

$0.36 \times 2 = 0.72$	0
$0.72 \times 2 = 1.44$	1
$0.44 \times 2 = 0.88$	0
$0.88 \times 2 = 1.76$	1
$0.76 \times 2 = 1.52$	1
$0.52 \times 2 = 1.04$	1
$0.04 \times 2 = 0.08$	

$(.61)_{10} \longrightarrow (.470)_8$

$0.61 \times 8 = 4.88$	4
$.88 \times 8 = 7.04$	7
$.04 \times 8 = 0.32$	0

$(.23)_{10} = (.3AE)_{16}$

$0.23 \times 16 = 3.68$	
$0.68 \times 16 = 10.88$	
$0.88 \times 16 = 14.08$	

Scanned by CamScanner

Representation of Binary Integer

1.) Unsigned Binary Integers
2.) Signed Binary Integers

1) **Unsigned:** There is no sign bit. All the bits are used to show magnitude of integer

ex- $(12)_{10} \rightarrow (1100)_2$

2) **Signed Binary:**

MSB (most significant Bit) used
0 → Positive sign
1 → Negative sign

Negative Integer:

These number can be identified by the MSB is 1. There are three possible ways ways with which negative Binary integer can be represented

1. Signed - Magnitude Repre. ex - 10001001
2. Signed - 1's complement Repre. → 11110111
3. Signed 2's complement Repre 11110111

Scanned by CamScanner

TNPL

Page No.
Date / /

compliment.

Diminished radix complement

$(N)_r$: contain n digit

then (r-1)'s complement → $(r^n - 1) - N$

Ex $(546700)_{10}$ 9's complement fi.

$N = 6$
$r = 10$

9's complement → $(10^6 - 1) - 546700$

$999999 - 546700$

$= 453299$

Ex $(1011000)_2$ is complement find

$N = 7, r = 2$ $(r^n - 1) - N$

$(2^7 - 1) - 1011000$

$= 10000000 - 1 - 1011000$

$= 1111111 - 1011000$

$= 0100111$

radix complement

Now r's complement $r^n - N$

Ex $N = 012398$ 10's complem

$(10)^6 - 012398$

$= 987602$

Ex $N = 1011000$, 2's compl

$2^7 - 1011000$

$10000000 - 1011000$

$= 0101000$

Scanned by CamScanner

Mean
S/c

Page No.
Date / /

[M−N]

Substration using (r-1)'s Complement

1. With m we add (r-1)'s complement & subtract
2. If $m \geq n$ there is a carry
   In this case we have to discard the carry also add 1 to erase -1
   $\Rightarrow M + (r^n - 1) - N$
   $\Rightarrow M - N + r^n - 1$

3. If $m < n$ then $((r^n - 1) - (N - M))$

   here N - 9's compl of m
   Ans also 9's complement
   take negative sign

Exe 72532 − 03250

9's compl of 03250 → 96749

```
  72532
+ 96749
 169281
   +1
 69282
```

discard carry

Answer

03250 − 72532

9's compl of 72532 → 27467

```
 03250
 27467
 30717
```

which 9's compl of ans

9's compl of 30717 → −69282

take negative sign

Scanned by CamScanner

Ex 1010100 − 1000011 using 1's complement

1's comp of 1000011 → 0111100

```
        1010100
      + 0111100
        10010000
discard carry 1
                +1
        0010001
```

Ans

Ex 1000011 − 1010100

1's comp of 1010100 → 0101011

```
        1000011
        0101011
        1101110
```

1's compl of 1101110 → −0010001 Ans

Substration using r's complement

1. Add the M to the r's complement of subtrahend
   → $M + (r^n - N)$
   ⇒ $M - N + r^n$

2. If $M \geq N$, the sum will produce carry which is discarded

3. If $M < N$, then $r^n - (N - M)$
   which is r's complement of N−M
   obtain ~~r's of com~~ the answer take r's compl. of sum and raise negative sign

Scanned by CamScanner

Page No
Date

(1) 72532 − 03250 (using 2's complement

10's compl. 03250 → 96750

→ 72532
96750
carry 1 69282

69282 Ans

(2) 03250 − 72532

10's compl. of 72532 → 27468

→ 03250
27468
30718

10's com. of 30718 → −69282
result −ve

Ex 1010100 − 1000011

2's com 1000011 → 0111101

: 1010100
0111101
1 0010001

ds carry

Ans 0010001

Ex 1 1000011 − 1010100

2's comp of 1010100 → 0101100

1000011
0101100
1101111

2's com of 1101111 → 0010000
+1
10001 Ans

Scanned by CamScanner

Different codes

Decimal	8421	2421	excess 3	84-2-1
	0000	0000	0011	0000
	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	0111	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1001
9	1001	1111	1100	1111

Addition BCD

```
   4        0100
+  8        1000
            ----
            1100   This is not in BCD, 8421
            0110   so add 6
            ----
   0001     0010
     1        2      -> 12
```

Ex

```
   184      0001 1000 0100
   576      0101 0111 0110
            ---------------
            0111 10000 1010   Not 1
                 0110  0110
                 ----  ----
                 0110  0000
                       carry
            760 A
```

1) Weighted code:

each position of the digit has some specific weight

Ex BCD

2) Non weighted codes:

Do not follow the positional weighting principle.

ex XS3 code, gray code.

③ Reflective code:

complement of each other

2421 and XS3 code are complement

Gray codes:

Binary → 1 1 0 1 0

Gray → 1 0 1 1 1

B(4) ~~Binary~~	Gray code	BCD	Gray
0000	0000	1000	1100
0001	0001	1001	1101
0010	0011	1010	1111
0011	0010	1011	1110
0100	0110	1100	1010
0101	0111	1101	1011
0110	0101	1110	1001
0111	0100	1111	1000

Gray 1 0 1 0 1 1

Binary 1 1 0 0 1 0

Use of gray code: ① used in optical encoder disks to reduce the possibility of error occuring due to shift of brushes

② Gray codes are also used in Karnaugh map method for minimisation of Boolean expression

Character codes

① ASCII code — 7 bit code
② EBCDIC code — 8 bit code
used in IBM

Error Detection and correction codes

(a) Parity code
(b) Check sums

parity code: extra bit is attached to the message. called parity bit.

Example odd parity transmission

Ex	Parity	Message
	1	1100101
	0	1010100
	1	1111000

ex even parity

	1	1011000
	0	1000001

(b) **Check sum:** Since parity method can not detect error in two bits.

In this method the transmitter transmits word and adds this word to the sum of all the words which.

Ex

Message 1 :	1010110
Message 2 :	1100100
Checksum	1011010

Scanned by CamScanner

Boolean Algebra

Boolean Algebra is an algebra that deals with binary variables and logic operations
There are boolean function separates by an algebraic expression consisting of binary variables. The constants 0 and 1 and the logical operation symbol
The basic logic operation are AND, OR, NOT

A	B	A-B	A+B
0	0	0	0
0	1	0	1

⇒ Boolean function: A Boolean function consisting of binary variables, the logic operation symbols, parenthesis and equal sign. For a given value of variables, the Boolean function can be either 0 or 1

$$F = x + y'z$$

Truth table:-

It is relationship between a function and its binary variable can be represented in the form of a truth table. To represent a function in a truth table, we need a list of 2^n combination of n binary variable

⇒ postulates and theorem.

1. $x + 0 = x$
   $x \cdot 1 = x$

2. $x + x' = 1$
   $x \cdot x' = 0$

Scanned by CamScanner

3. $x + x = x$ (Identity law)

$x \cdot x = x$

proof: (i) $x + x = (x+x) \cdot 1$ [$\because x \cdot 1 = x$

$= (x+x)(x+x')$ [$x + x' = 1$]

$= x + x \cdot x'$

$= x$ [$\because x \cdot x' = 0$

(ii) $x \cdot x = x \cdot x + 0$

$= x \cdot x + x \cdot x'$ [$x \cdot x' = 0$

$= x(x + x')$

$= x$

4. (i) $x + 1 = 1$

(ii) $x \cdot 0 = 0$

$x + 1 = 1 \cdot (x+1)$ [$1 \cdot x = x$

$(x + x')(x+1)$

$= x + x' \cdot 1$

$= x + x'$

$= 1$

(ii) $x \cdot 0 = 0$ (by duality

(5) $(x')' = x$

$x + xy = x$

(6) $x(x+y) = x$ [Absorption Theorem

proof:

(i) $x + xy = x \cdot 1 + xy$

$= x(1+y)$

$= x \cdot 1$ [$\because x \cdot 1 = x$

$= x$

(ii) $x(x+y) = x$ [by duality

$\Rightarrow x + xy$

$\Rightarrow x + xy$

$\Rightarrow x(1+y) = x$

Scanned by CamScanner

7) $x+y = y+x$ [Commutative law]

$x \cdot y = y \cdot x$

8) $x+(y+z) = (x+y)+z$ [Associative law]

$x(yz) = (xy)z$

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

$x+yz = (x+y)\cdot(x+z)$ (Distributive law

10) $xy+x\bar{y} = x$

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

11 $x+xy = x$ [Reductidence law

$x(x+y) = x$

12 $x+x'y = x+y$

$x(x'+y) = xy$

13 $(x+y)' = x'y'$ (De-morgan's law)

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

Minterms:

A binary variable may appear either in its normal form x or in complement form x' consider two binary variable x and y combined with AND operation. Thus there are four possible combination $x'y'$; $x'y$; xy' and xy Each of these four AND terms is called a min term; or a standard product

For 'n' variable; 2^n min terms can be obtained

Min terms are relevant to L

→ Maxterms: For two x and y combined with OR operation.

Thus $x+y$, $x+y'$, $x'+y'$, is called a max terms, or a standard sum

convert following truth table into boolean function

Ex

x	y	z	F	Min terms	Max terms
0	0	0	1	$\bar{x}\bar{y}\bar{z} = m_0$	$x+y+z = M_0$
0	0	1	0	$\bar{x}\bar{y}z = m_1$	$x+y+\bar{z} = M_1$
0	1	0	0	$\bar{x}y\bar{z} = m_2$	$x+\bar{y}+z = M_2$
0	1	1	0	$\bar{x}yz = m_3$	$x+\bar{y}+\bar{z} = M_3$
1	0	0	1	$x\bar{y}\bar{z} = m_4$	$\bar{x}+y+z = M_4$
1	0	1	1	$x\bar{y}z = m_5$	$\bar{x}+y+\bar{z} = M_5$
1	1	0	1	$xy\bar{z} = m_6$	$\bar{x}+\bar{y}+z = M_6$
1	1	1	0	$xyz = m_7$	$\bar{x}+\bar{y}+\bar{z} = M_7$

Soln

Take sum of minterms corresponding to 1

$$F = \bar{x}\bar{y}\bar{z} + x\bar{y}\bar{z} + x\bar{y}z + xy\bar{z}$$
$$= m_0 + m_4 + m_5 + m_6$$
$$= \Sigma(m_0, m_4, m_5, m_6)$$

OR

Take product of maxterms corresponding to 0 values

$$F = (x+y+\bar{z})(x+\bar{y}+z)(x+\bar{y}+\bar{z})(\bar{x}+\bar{y}+\bar{z})$$
$$= M_1 M_2 M_3 M_7$$
$$= \Pi(M_1, M_2, M_3, M_7)$$

Scanned by CamScanner

canonical form: If in a boolean function all the variable are appearing in each term, then it is said to be in canonical form.

Ex- $xyz + xy\bar{z}$ is canonical form

=> conversion from standard form to canonical form in sum of product (Min terms

$F = x + y\bar{z}$

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

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

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

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

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

$m_7 + m_6 + m_5 + m_4 + m_2$

$\Sigma(2, 4, 5, 6, 7)$

=> Conversion from standard form to canonical form in product of sums

$F = xy + \bar{x}z$

$= (x+\bar{x})(xy+z)$ (distributive)

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

$= (\bar{x}+y)(x+z)(y+z)$ $\because x+\bar{x} = 1$

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

introducing missing term

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

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

$= M_0 M_2 M_4 M_5$

$= \pi(0, 2, 4, 5)$

Scanned by CamScanner

→ conversion from canonical from in product of sum up to simple form in sum of product

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

$= (x+y)(x+\bar{y})(\bar{x}+\bar{y}+z)$ $[\because (A+B)(A+\bar{B}) = A]$

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

$= x\bar{x}+x\bar{y}+xz$

$= x\bar{y}+xz$

ex 1) $x+\bar{x}y+\bar{y}+(x+\bar{y})\bar{x}y$

$= x+y+\bar{y}+x\bar{x}y+\bar{x}y\bar{y}$ $[\because A+\bar{A}B = A+B$

$= x+y+\bar{y}+0+0$

$= x+1$ $[\because A+\bar{A}=1$

$= 1$ $[A+1=1$

2) $z(y+z)(x+y+z)$

$(zy+zz)(x+y+z)$ $[A.A = A$

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

$z(x+y+z)$

$= zx+zy+zz$

$= zx+zy+z$

$= zx+z(y+1)$

$= zx+z$

$= z(x+1)$

$= z$

Scanned by CamScanner

③ $(x+y)(\bar{x}+z)(y+z)$

$= (\underset{0}{x\bar{x}} + xz + \bar{x}y + yz)(y+z)$

$= xyz + yyz + \bar{x}yy + xzz + \bar{x}yz + yzz$

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

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

$= yz + zx + zy + zy + \bar{x}y$

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

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

$= yz + zx + zy + \bar{x}y$

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

$= y\bar{x} + yz + xz$

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

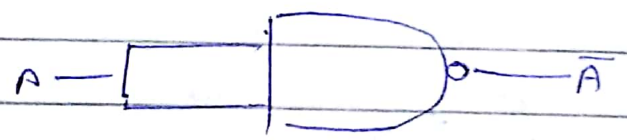
$= xz + y\bar{x} + xyz + yz\bar{x}$

$= zx(1+y) + \bar{x}y(1+z)$

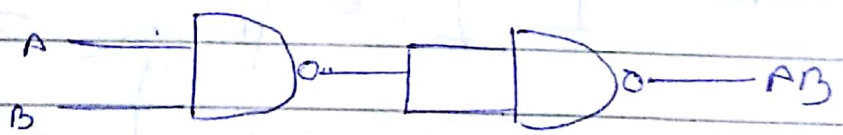
$= zx + \bar{x}y$

Realization of other function using NAND gate

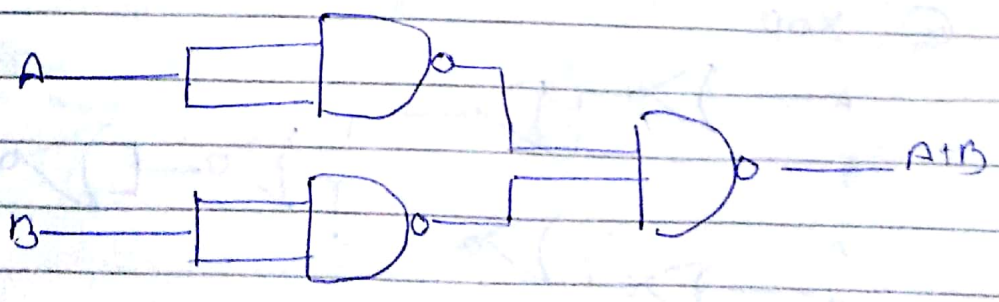
① NOT



② AND

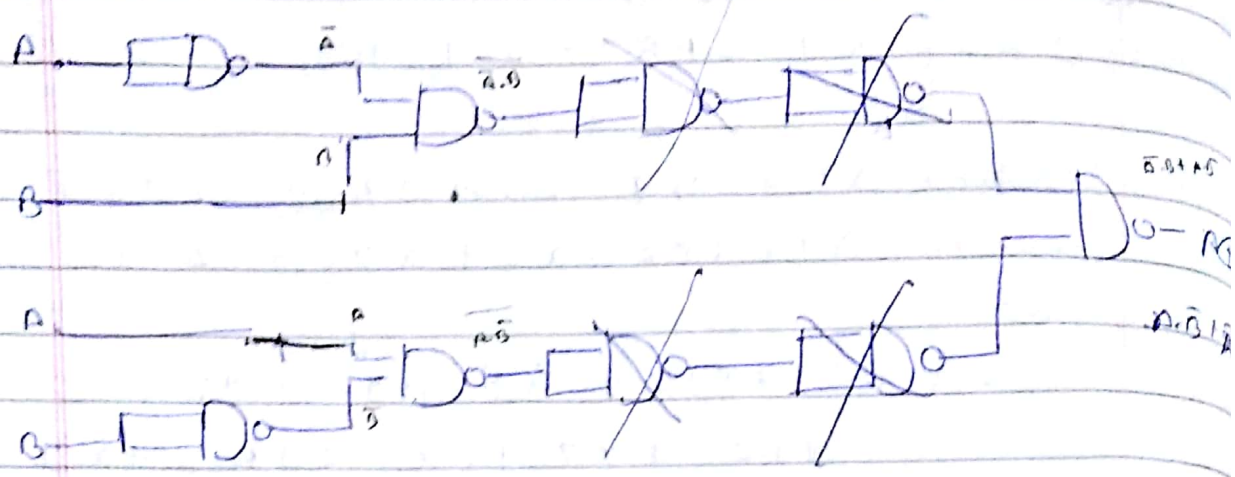


③ OR



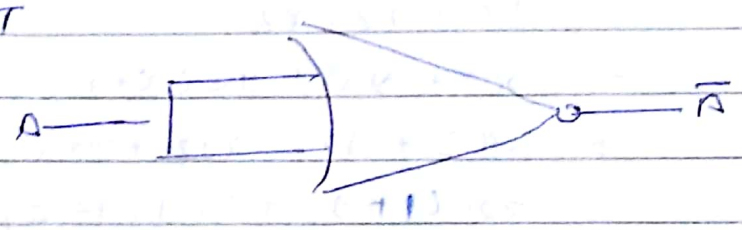
Scanned by CamScanner

④ XOR -

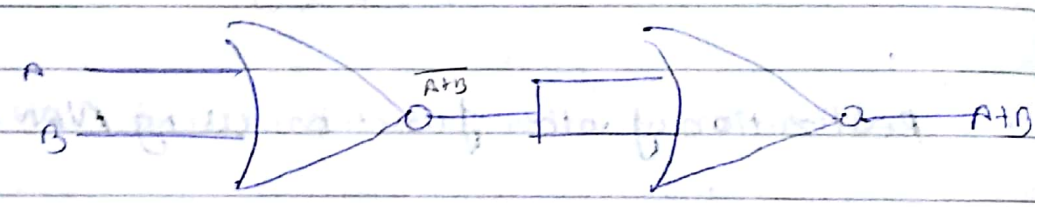


→ Realisation of other function using NOR gate

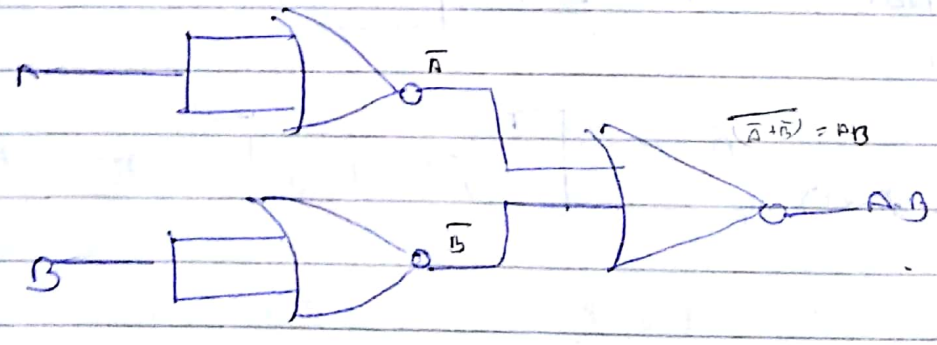
1) NOT



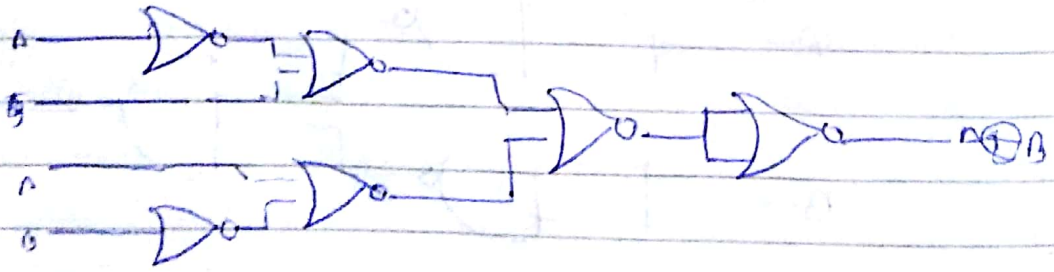
② OR



③ AND



④ XOR



Scanned by CamScanner

Map simplification:

The map method provides a simple, straightforward procedure for simplifying Boolean expression. This method may be regarded as a pictorial arrangement of the truth table which allows an easy interpretation for choosing the minimum number of terms needed to express the function algebraically.

The method is also known as the Karnaugh map or K-map

The map is a diagram made up of square, with each square representing one minterm. The square corresponding to the minterms that produce 1 for the function are marked by a 1 and the other marked by 0.

The map for function of 2, 3, and 4 variables

	$\bar{A}$	A
$\bar{B}$	$\bar{A}\bar{B}$ m_0	$A\bar{B}$ m_2
B	$\bar{A}B$ m_1	AB m_3

	00	01	11	10
0	$\bar{x}\bar{y}\bar{z}$ m_0	$\bar{x}\bar{y}z$ m_1	$\bar{x}yz$ m_3	$\bar{x}y\bar{z}$ m_2
1	$x\bar{y}\bar{z}$ m_4	$x\bar{y}z$ m_5	xyz m_7	$xy\bar{z}$ m_6

y

z

x

	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$ m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	$\bar{A}B\bar{C}D$ m_{13}	$ABCD$ m_{15}	$ABC\bar{D}$ m_{14}
10	$A\bar{B}\bar{C}\bar{D}$ m_8	$A\bar{B}\bar{C}D$ m_9	$A\bar{B}CD$ m_{11}	m_{10}

Scanned by CamScanner

Q Simplify $F = \Sigma(0,1,2,4)$

	$\bar{y}\bar{z}$ (00)	$\bar{y}z$ (01)	yz (11)	$y\bar{z}$ (10)
$\bar{x}$ 0	1 (0)	1 (1)	(3)	1 (2)
x 1	1 (4)	(5)	(7)	(6)

$F = \bar{x}\bar{y} + \bar{y}\bar{z} + \bar{x}\bar{z}$

As 3 group are formed so there are 3 product in answer

Also $F = m_0 + m_1 + m_2 + m_4$

Q Realization of product of sum expression from K-map

$f(A,B,C,D) = \Sigma(0,1,2,4,7,8,9)$

$= m_0 + m_1 + m_2 + m_4 + m_7 + m_8 + m_9$

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

$F' = \bar{B}CD + AB + B\bar{C}D + AC + BC\bar{D}$

$AB + AC + B\bar{C}D + \bar{B}CD + BC\bar{D}$

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

Scanned by CamScanner

Don't care condition.

The 1's and 0's in the map represents the minterms that make the function equal to 1 or 0. There are occassions when it does not matter if the function produce 0 or 1 for a given minterm. This are called 'don't care conditions' and marked with a X or d in the map

Ex

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	d
1	0	0	d
1	0	1	0
1	1	0	0
1	1	1	0

$\therefore \Sigma m(1,2) + \Sigma d(3,4)$

	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	0 0	1 1	d 3	1 12
x	d 4	0 5	0 7	0 6

(1) Make group of 1's & d's.

$F = x\bar{y}\bar{z} + \bar{x}z + \bar{x}y$

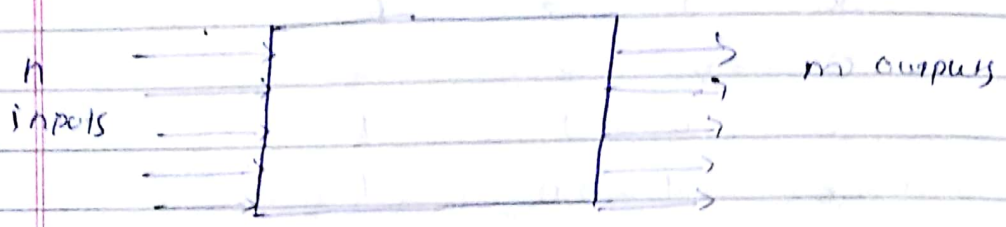
(2) By make group of 0's and d's

$\bar{F} = \bar{y}\bar{z} + x$,

Scanned by CamScanner

combinational circuit:

A combinational circuit consist of logic gates whose output at any time are determined from only the present combination of input. A combinational circuit perform an operation that can be specified logically by a set of boolean functions



For 'n' input variable, there are 2^n possible binary input combination, for each possible input combination, there is one possible value

1) Half Adder:

A combinational circuit that perform the addition of two bits is called a half adder. This circuit needs two binary inputs and two binary outputs.

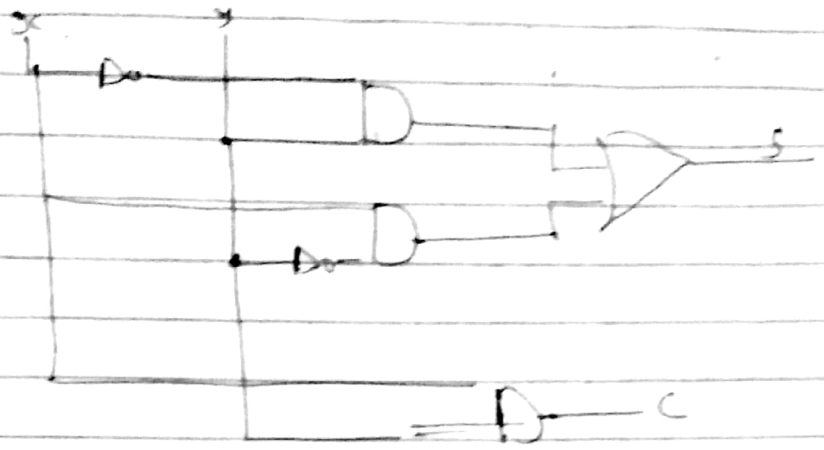
x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$S = x'y + xy' = x \oplus y$

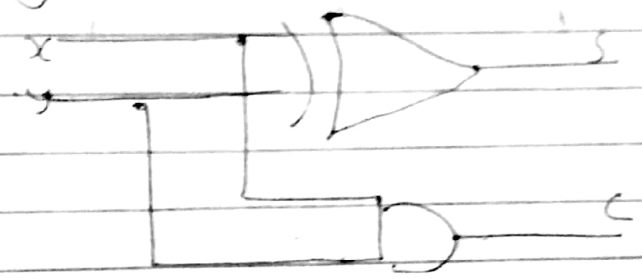
$c = xy$

Scanned by CamScanner

The logic diagram of half adder in sum of product.



Using XOR gate

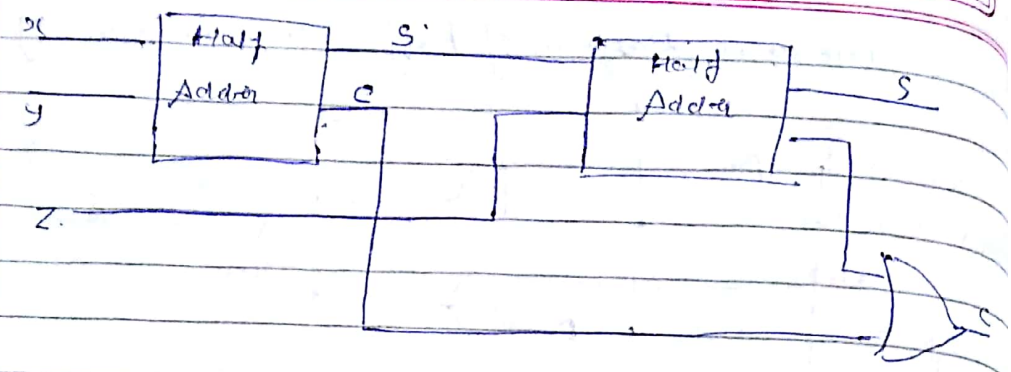


The limitation of half adder is that it can add only 2 bit at a time

(2) Full Adder: It is a combinational circuit that forms the arithmatic sum of 3 bits. It consist of 3 inputs and 2 outputs. Two of the input variable x and y represent the two significant bits to be added. The third input z represent the carry from the previous lower significant position

x	y	z	c	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

Scanned by CamScanner



The Boolean function

	$\bar{y}\bar{z}$	$y\bar{z}$	yz	$\bar{y}z$
$\bar{x}$	0	1	0	1
x	1	0	1	0

sum

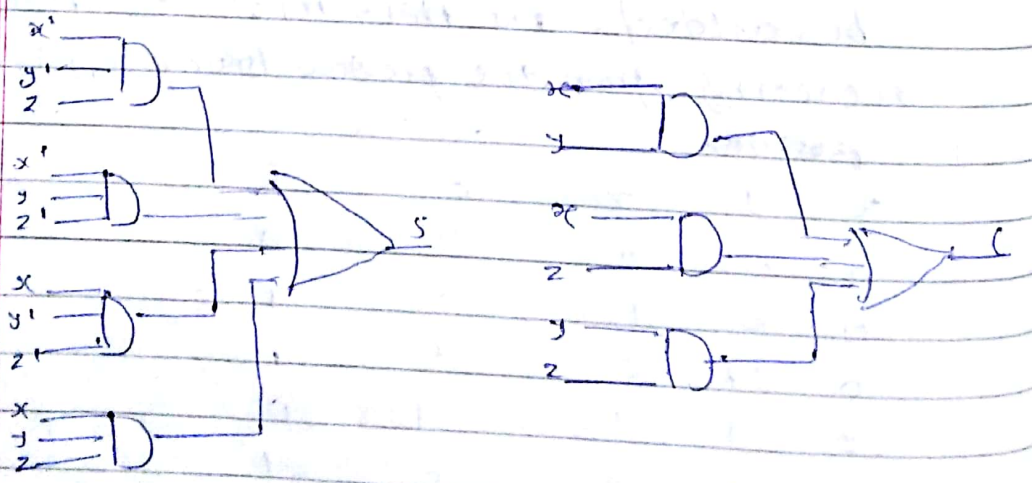
	$\bar{y}\bar{z}$	$y\bar{z}$	yz	$\bar{y}z$
$\bar{x}$	0	0	1	0
x	0	1	1	1

carry

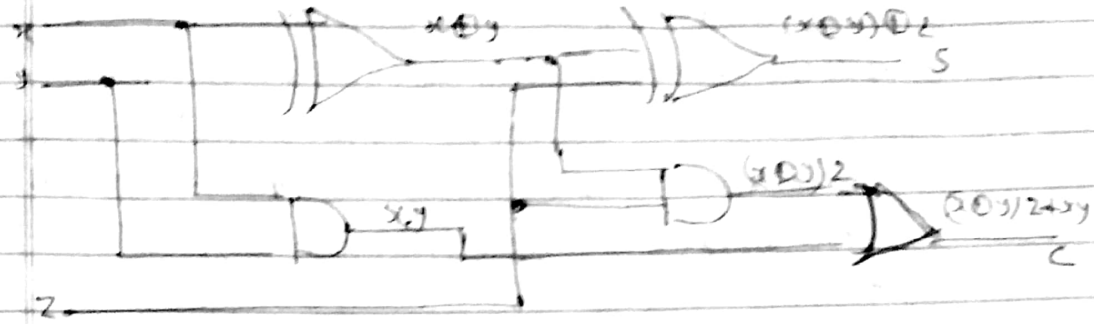
$$S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$$
$$= \bar{x}(\bar{y}z + y\bar{z}) + x(\bar{y}\bar{z} + yz)$$
$$= x \oplus y \oplus z$$

$$C = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$$
$$= xy + xz + yz$$
$$= (x \oplus y)z + xy$$

<u>Using</u> sum of product



using two half adder

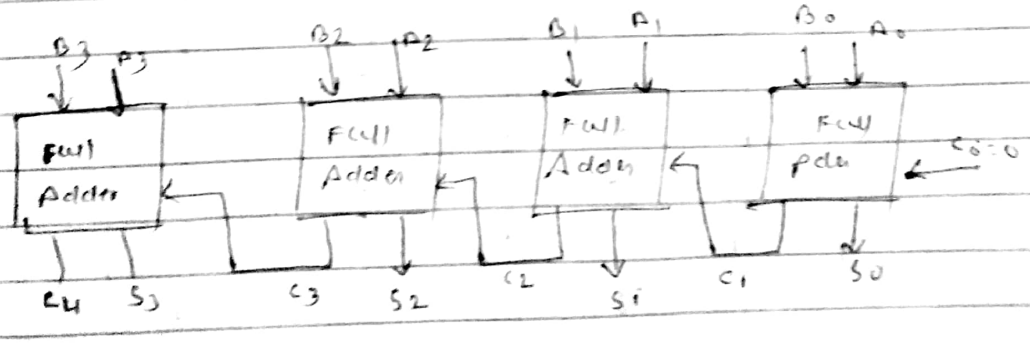


Binary Adder:

A Binary Adder is a digital circuit that produces the arithmatic sum of two binary numbers. It can be constructed with full Adder connected in cascade, with the output carry from each full adder connected to the input carry of the next full adder in the chain.

Consider two binary number A = 1011 and B = 0011. Their sum 1110 is as

Input carry	0	1	1	0	C_i
Augend	1	0	1	1	A_i
Addend	0	0	1	1	B_i
sum	1	1	1	0	S_i
output carry	0	0	1	1	C_{i+1}



Result ⇒ $C_4 S_3 S_2 S_1 S_0$

Page No.:
Date: / /

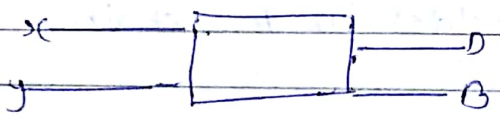
4. **Half Substractor:**

A combinational circuit that perform the subtraction of two bits is called half substractor. This circuit needs two binary input and two binary output. The input are the values to be subtracted. The output variable give the difference and borrow.

x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$\therefore D = x \oplus y$, $B = \bar{x}y$

Block diagram



(5) **Full Substractor:**

It is a combinational circuit that gives the difference of 3 bits. It consist of 3 input and 2 output. The outputs represent the difference and borrow

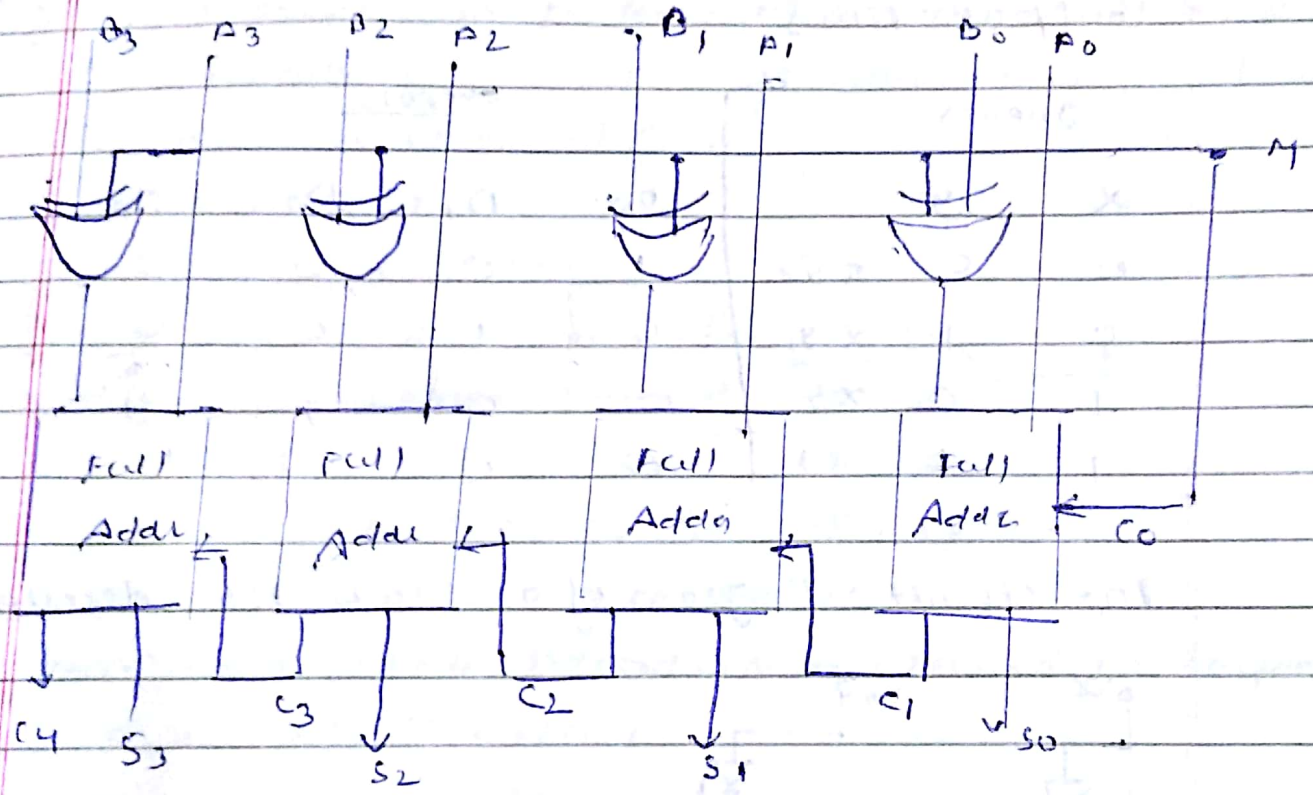
x	y	z	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$\therefore D = A \oplus B \oplus C$, $B = \bar{x}z + \bar{x}y + yz$

Scanned by CamScanner

Binary Adder-Substractor:

The addition and substraction operation can be combined into one circuit by including a XOR gate with each full adder. A 4-bit adder-substractor is as follows



The mode input M control the operation when M=0, the circuit is an adder when M=1, the circuit become a substractor. There is delay in availability of carry to the subsequent full-adder. To avoid this, we generate carry separately and give it directly to the full adders. This is called, "carry look ahead Method"

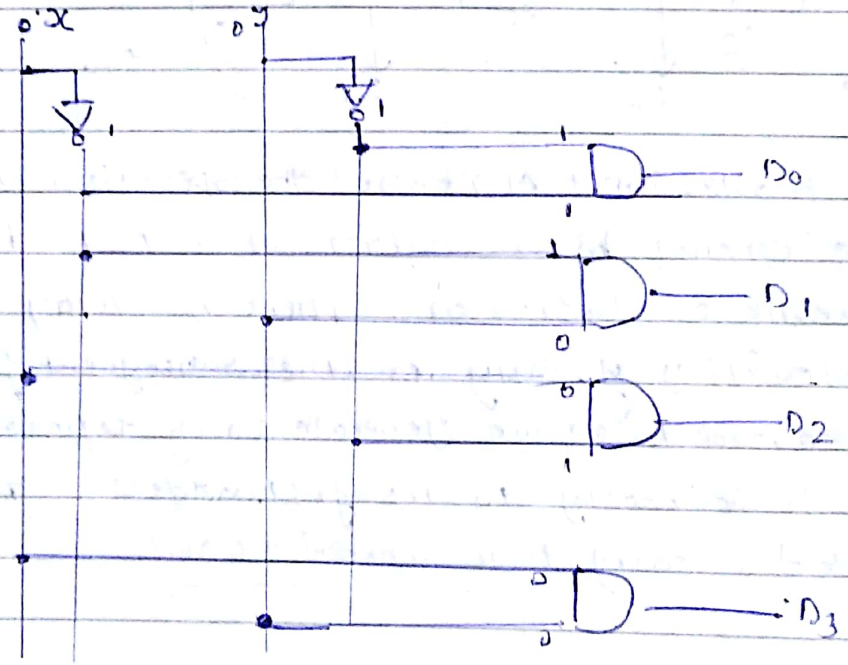
Scanned by CamScanner

(7) Decoder:

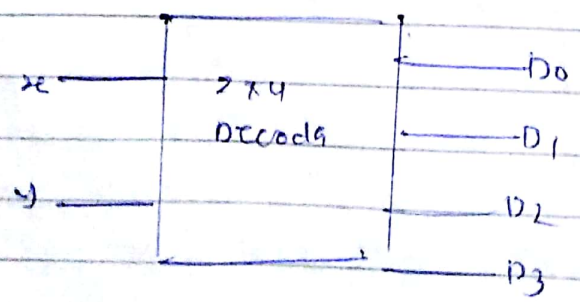
A binary code of 'n' bits is capable of representing up to 2^n distinct element of coded information. A decoder is a combination circuit that converts binary information from 'n' input lines to a maximum of 2^n unique output lines.

Inputs			Outputs			
x	y		D_0	D_1	D_2	D_3
0	0	$x\bar{y}$	1	0	0	0
0	1	xy	0	1	0	0
1	0	$x\bar{y}$	0	0	1	0
1	1	xy	0	0	0	1

The circuit diagram of a 2 to 4 line decoder is



The Block diagram of 2 to 4 line decoder is as follo



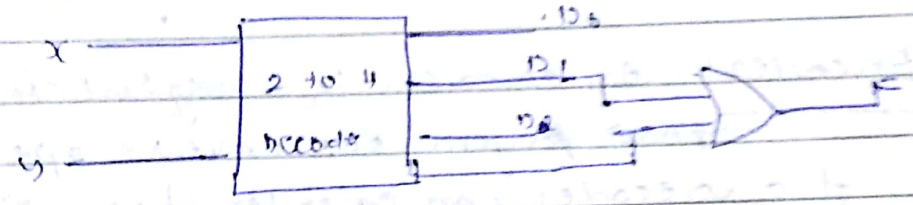
Scanned by CamScanner

(*) Decoder is also known as minterm generator because $D_0 = \bar{x}\bar{y}$, $D_1 = \bar{x}y$, $D_2 = x\bar{y}$, $D_3 = xy$ are minterms

(*) If there are 'n' variable we use "$n \times 2^n$" decoder

Q Is it possible to implement $F = xy + \bar{x}y$ by decoder

Since decoder is a minterm generator and the function is also in canonical form, so, it can be implemented.



$F = xy + \bar{x}y = \Sigma(1,3)$

Q How many 2 to 4 Decoders are needed to implement 5 to 32 Decoder?

S. No. of required decoders $= \frac{32}{4} = 8$

$\therefore$ 8 · 2 to 4 Decoder are required

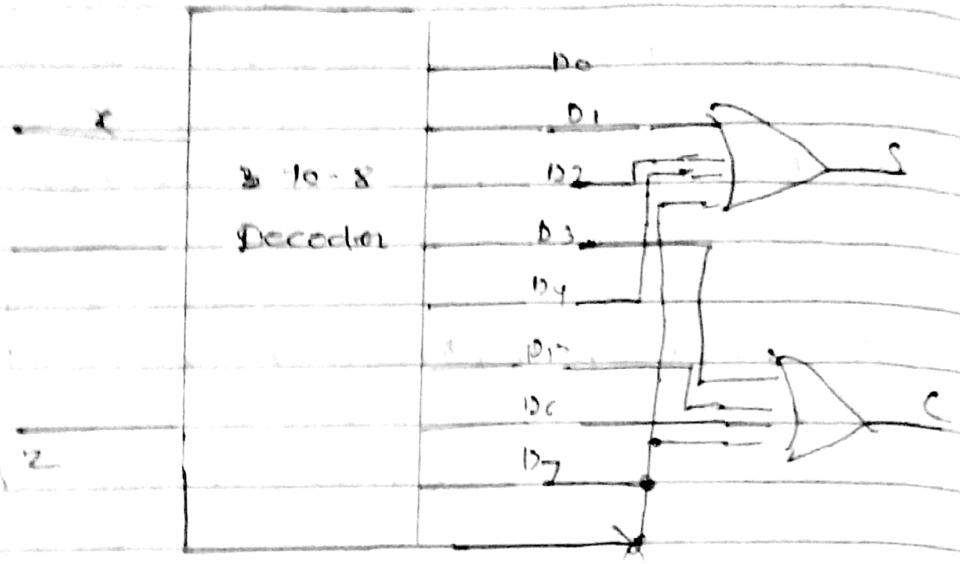
(*) Full Adder can also be implemented using a decoder as follows

$S(x,y,z) = \Sigma(1,2,4,7)$

$C(x,y,z) = \Sigma(3,5,6,7)$

We will use 3-to-8 decoder as there are 3 variables

Scanned by CamScanner



(8) **Encoder:** An encoder is a digital circuit that perform the inverse operation of a decoder. An encoder has 2^n input ling and n output lines. The output lines generate the binary code corresponding to the input value.

An octal to binary encoder has 8 inputs and 3 outputs. only one input has a value of 1 at given time.

Truth table

E_7	E_6	E_5	E_4	E_3	E_2	E_1	E_0	x	y	z
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

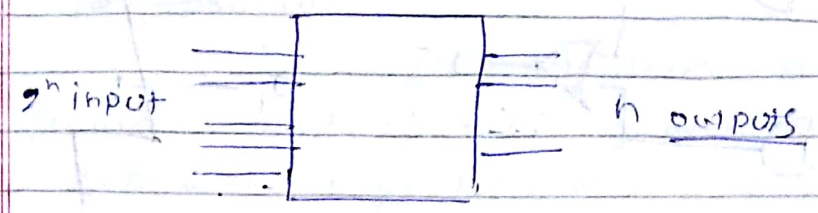
$x = E_4 + E_5 + E_6 + E_7$

$y = E_2 + E_3 + E_6 + E_7$

$z = E_1 + E_3 + E_5 + E_7$

Scanned by CamScanner

The block diagram



(*) The limitation of such encoder is that only one input can be active at any given time. If two input are active simultaneously, the output gives an undefined combination

(10) **De-multiplexer:**

Decoder with enable input is called Demultiplexer

The truth table for 1-to 4 demultiplexer is as follows

S_0	S_1	output
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

The circuit diagram for 1-to-4 Demultiplexer as

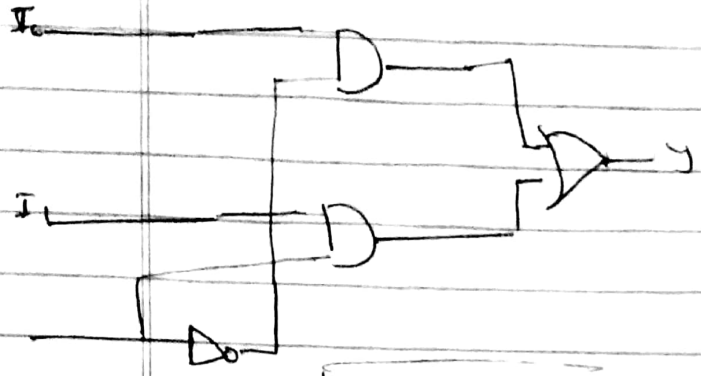
Multiplexer:

A multiplexer is a combinational circuit that select binary information from one of many input lines and direct it to a single output line. The selection of a particular input line is controlled by a set of selection lines

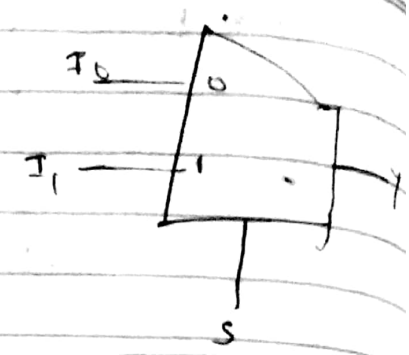
Normally there are 2^n input lines and n selection lines whose bit combination determine which input is selected.

A multiplexer is also called data selector.

Scanned by CamScanner

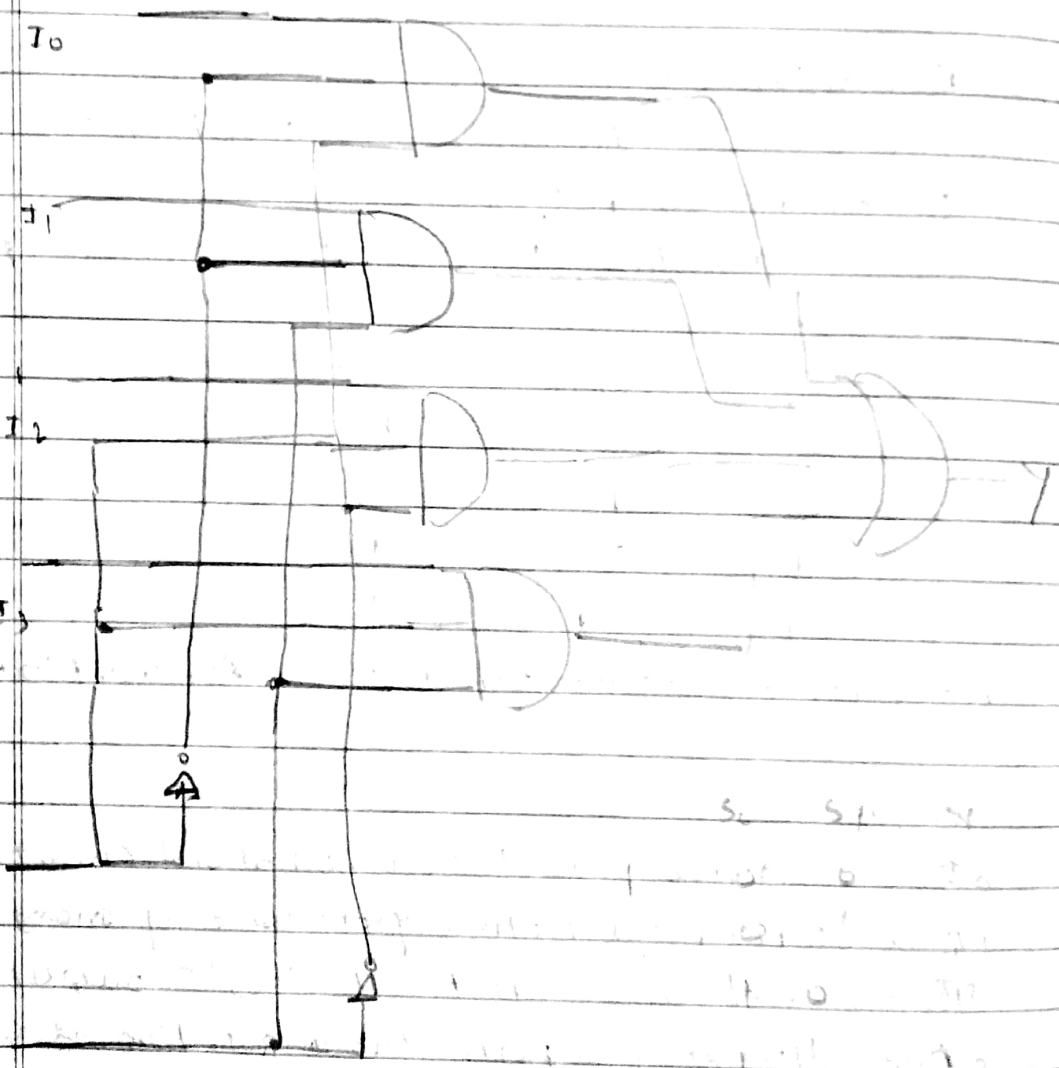


Logic diagram



Block diagram

Two 10 one line multiplexer

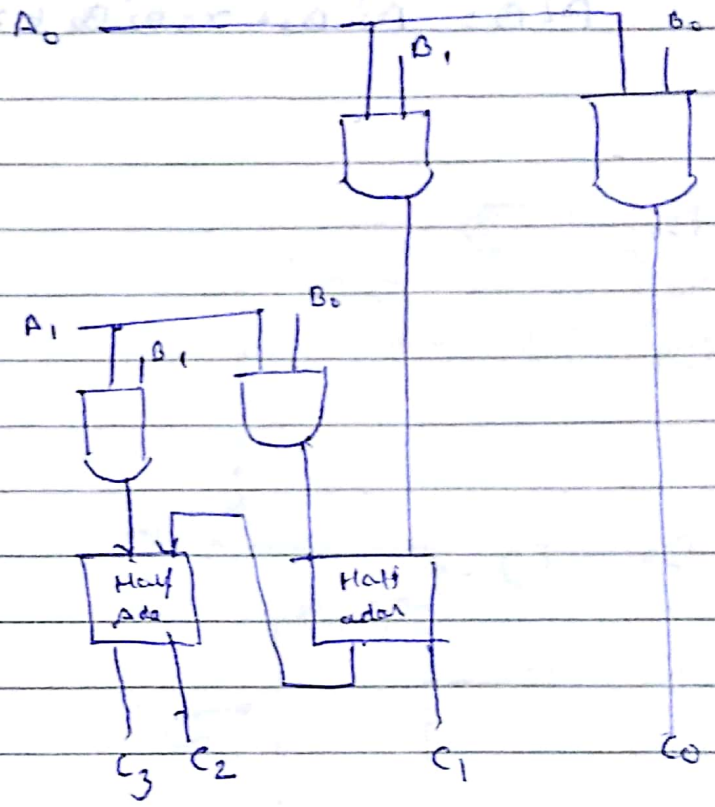


~~Multiplexer~~

Binary multiplier

The multiplication of two bits such as A_0 and B_0 produces a 1 if both bits are 1. Otherwise, it produces a 0. This is identical to an AND operation.

$$
\begin{array}{cccc}
 & & B_1 & B_0 \\
 & & A_1 & A_0 \\
\hline
 & & A_0B_1 & A_0B_0 \\
 & A_1B_1 & A_1B_0 & \\
\hline
C_3 & C_2 & C_1 & C_0
\end{array}
$$



Magnitude comparator

When the number are binary the digits are either 1 or 0. and the equality of each pair of bit can be expressed logically with an exclusive NOR function as

$$x_i = A_iB_i + \bar{A}_i\bar{B}_i$$

where $x_i = 1$ only if the pair of bit are equal. For equality to exist, all x_i variable must be equal to 1, a condition that directes an AND operation of all variables

$$A=B = x_3x_2x_1x_0$$

Scanned by CamScanner

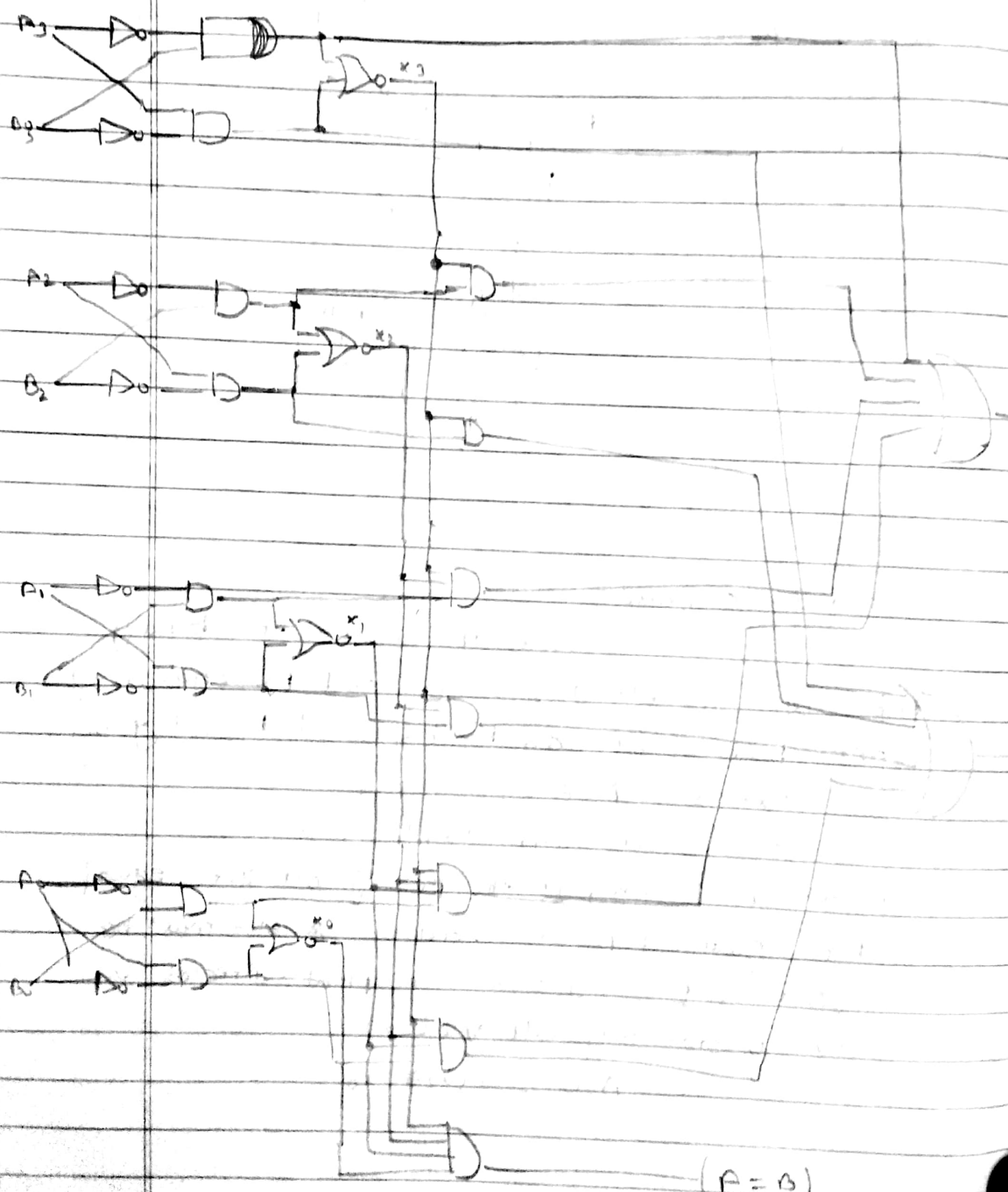
Page No.
Date / /

Now $A>B$ if the corresponding digit of A is 1 and B is 0.

$B>A$ if the corresponding A is 0 and B is 1

Now $A>B = A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0'B_0$

$A<B = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1 + x_3x_2x_1A_0'B_0$

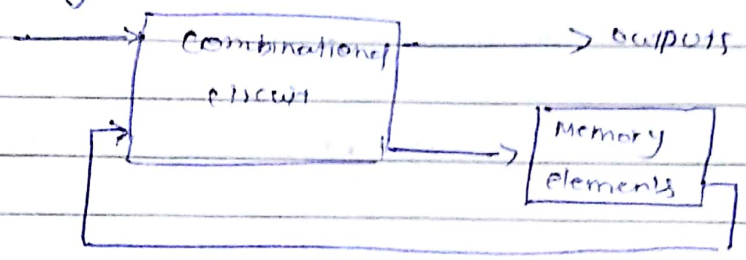


Scanned by CamScanner

sequential circuits

(1) Sequential circuit employ strong element in addition to logic gates. Their outputs are a function of the inputs and the state of the storage element. Thus the outputs of a sequential circuit depend not only on present value of input; but also on past input.

Block diagram.



The memory element are devices capable of storing binary information

There are two main types of sequential circuit
(1) Synchronous.
(2) Asychronous

Synchronous: A synchronous sequential circuit is a system whose behavious can be defined from the knowledge of its signals at discrete instants of time. Synchronization is achieved by a timing device called a 'clock generator' which provide clock signal.

Asynchronous:

The behavious of an asynchronous sequential circuit depends upon the input signals at any instant of time and the order in which the input changes

- The storage element used in clocked sequential circuit are called flip-flops

- [Storage elements that operate with signal levels are called latches, those controlled by a clock transitions are called flip-flops]

- Latches are said to be level sensitive devices. flip-flops are edge sensitive devices.

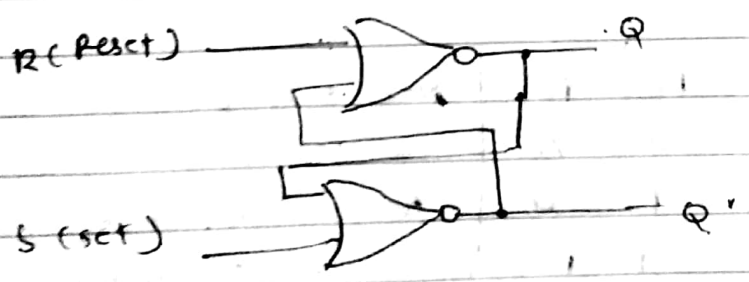
Latches

SR latches

The SR latch is a circuit with two cross coupled NOR gates or two cross coupled NAND and two input labeled 'S' for set and R for reset

when output Q = 1 and Q' = 0 latch is said to be in set state.

when Q = 0 and Q' = 1 it is in reset set.



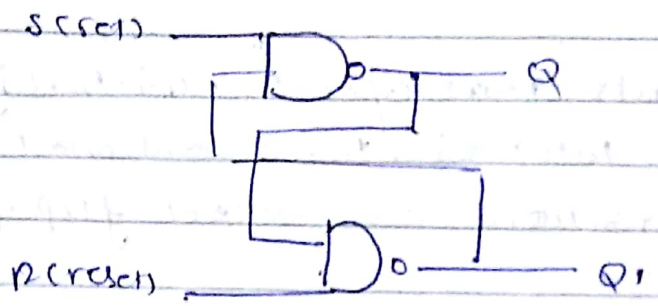
(Logic diagram)

S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

function table

SR latch using NOR gate

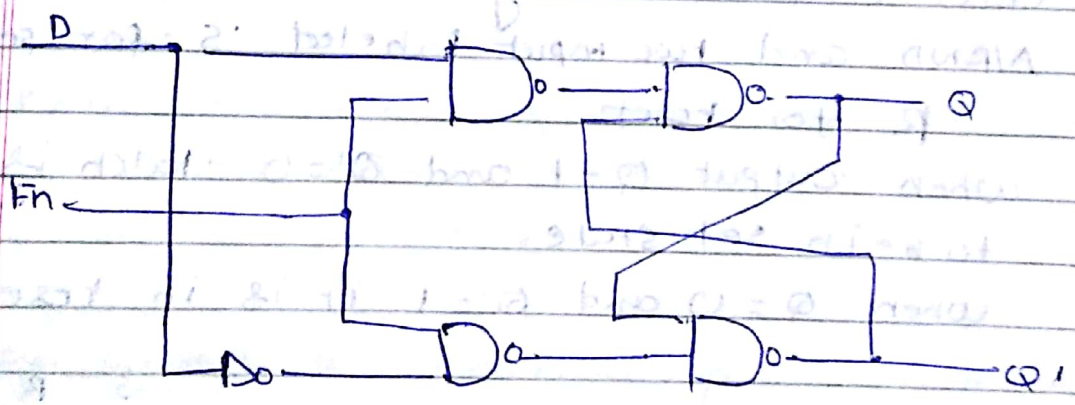
SR Latch with NAND gate



S	R	Q	Q'
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1 (Tab

D Latch:

One way to eliminate the undesirable condition of the indeterminate state in the SR latch is to ensure S and R are never equal to 1 at the same time. This is done in the D latch.



En	D	Next state of Q
0	X	No change
1	0	Q = 0 reset state
1	1	Q = 1 set value

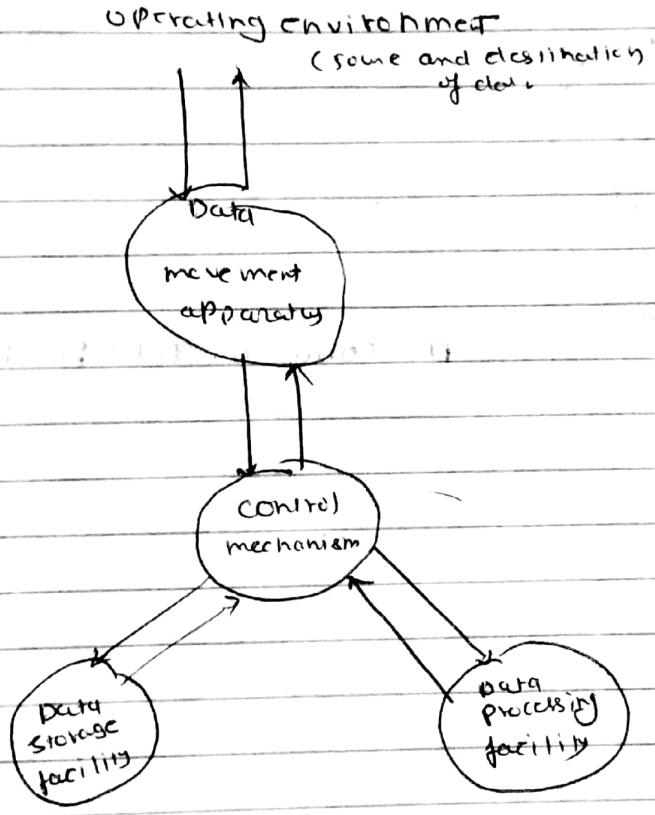
Scanned by CamScanner

Computer organization and stru

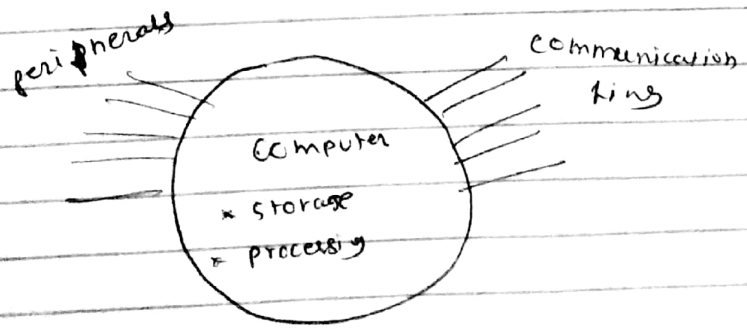
Structure and function:

(1) **Structure:** The way in which the component are interrelated

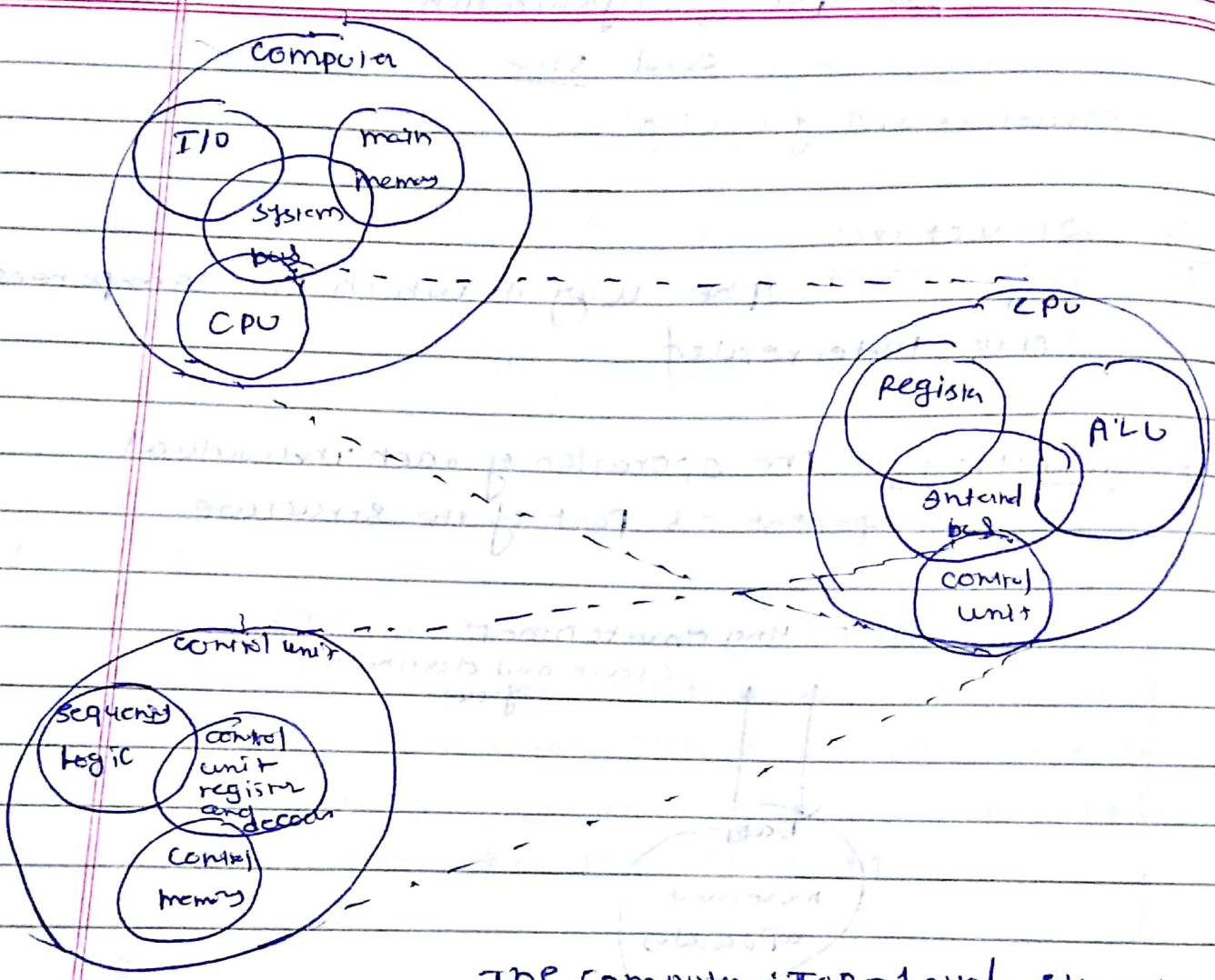
(2) **Function:** The operation of each individual component as part of the structure



A function view of computer



Structure the computer



The computer : Top-level structure

Scanned by CamScanner

Computer Memory system Overview

Location: Internal (eg. processor, register, main memory, cache) External (eg. optical, magnetic tape)	**performance** Access time Cycle time transfer time
Capacity: Number of words Number of bytes	**Physical type:** semiconductor Magnetic optical Magneto-optical
Unit of transfer: word Block	**Physical characteristics** volatile / non-volatile erasable / non-erasable
Access Method: Sequential Direct random Associative	**Organization** Memory modules

Key point Memory

Access time (latency):- For random access memory this is the time it takes to perform a read or write operation

Memory cycle time: It consists of access time plus any additional time required before a second access can commence.

transfer rate:

This is the rate at which data can be transferred into or out of a memory unit

for random access memory $= \frac{1}{(\text{cycle time})}$

Scanned by CamScanner

$\Rightarrow \quad T_N = T_A + \frac{n}{R}$

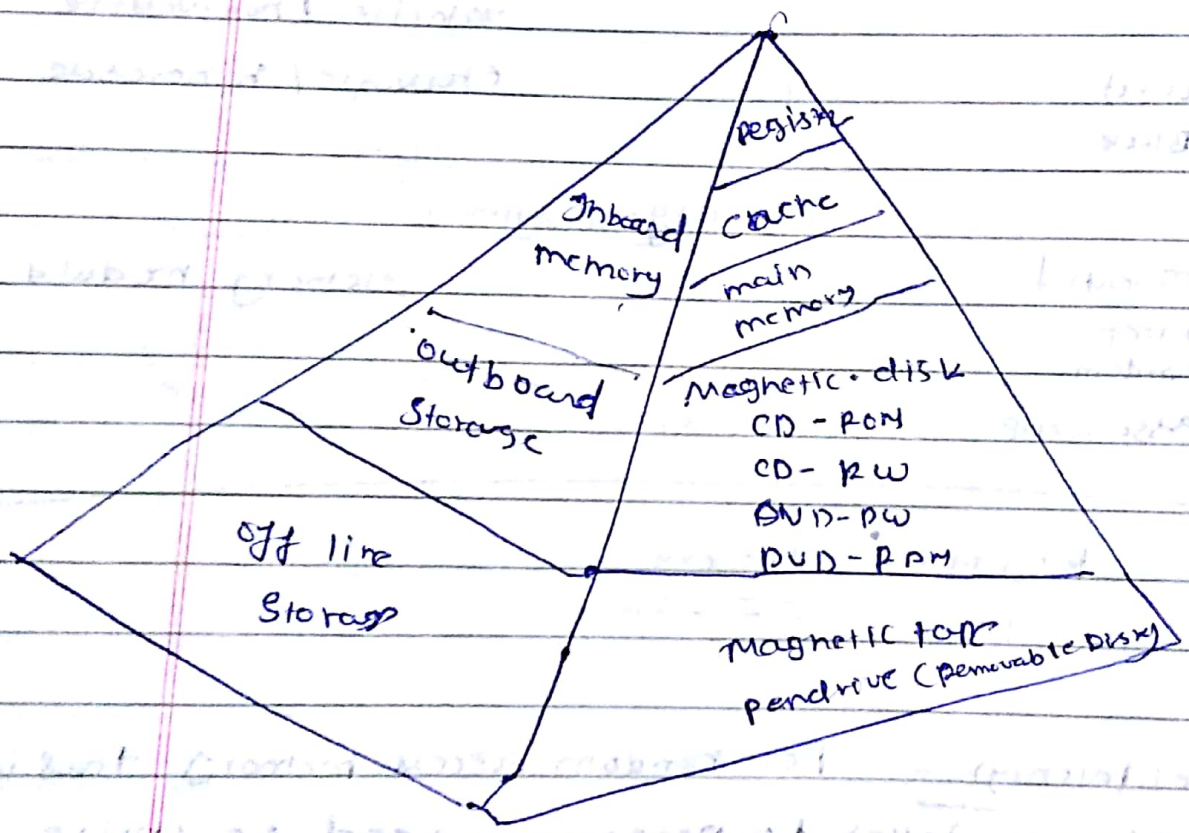
where T_N = Average time to read or write N bits

T_A = Average access time

n = number of bits

R = transfer rate

Memory Hierarchy :=>



Scanned by CamScanner

compare SRAM and DRAM

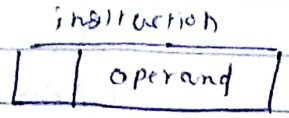
	SRAM	DRAM
① Definition	It is type of RAM SRAM essentially uses latches to store charge	also type of RAM use of capacitor to store bits in the form of charge
speed	faster	slower
size	bigger	smaller
cost	More expensive per bit	less expensive
capacity	comparative less	comparative more 5-10 time than SRAM
Access	easy	Harder
power consumption	less	more
Density	low density memory per chip	Hig density per chip
Refresh	No need to refresh	Need of refresh
②	use Bipolar or MOS technology	use MOSFET technology

Addressing

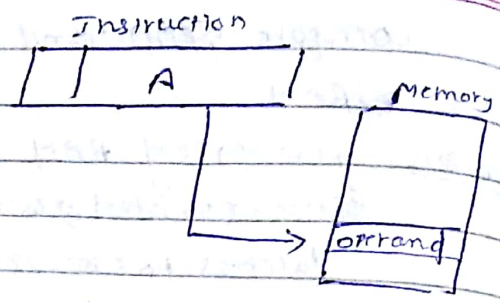
We would like to able to reference a large range of location in main memory. To acheive this objective, a variety of addressing technique are used

① Immediate
② Direct
③ Indirect
④ Register
⑤ Register indirect
⑥ Displacement
⑦ Stack

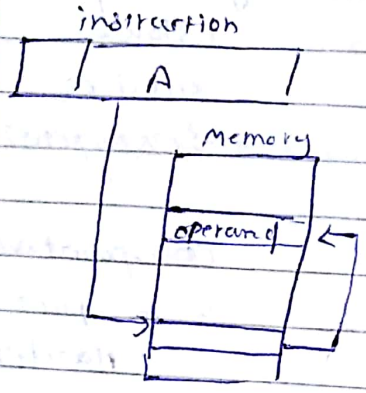
Scanned by CamScanner



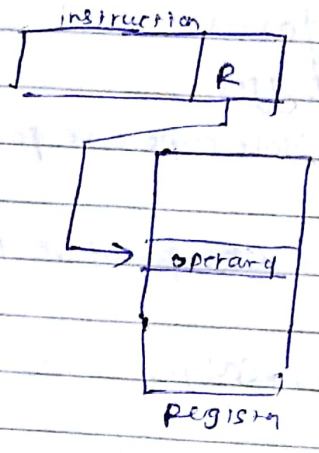
(a) Immidiate



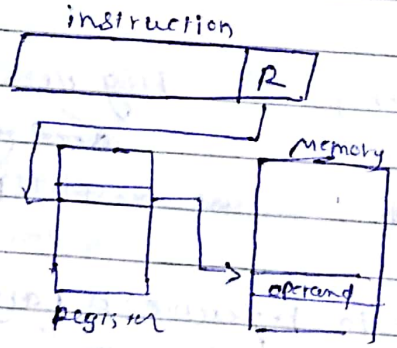
(b) Direct



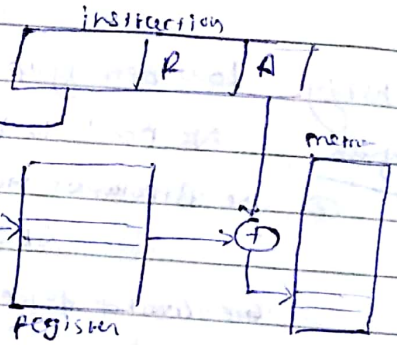
(c) Indirect



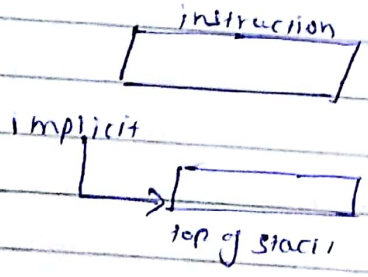
(d) Register



(e) Register indirect



(f) Displacement



(g) Stack

Note

A = content of an address field in the instruction

EA = effective address of the location containing the referenced operand

X = content of memory location X or register X

Immediate Addressing

simplest form of addressing is immediate addressing, in which the operand value is present in the instruction

operand = A

Use: to define and use constant values of variable

Advantage :- immediate addressing, no memory reference saving memory

Disadvantage :- size of the number is restricted

Direct Addressing

simple method, in which the address field contains the effective address of operand

EA = A

Used: in earlier generation of computer

Disadvantage : Provide only a limited address space

Indirect Address

with direct addressing, the length of address field is usually less than the word length thus limiting address range. one solution is to have the address field refer to the address of a word in memory, which in turn contains a full length address of the operand. this is known as indirect Addressing

EA = (A)

parenthesis represent contents

Advantage : word length of N, an address space of 2^N available

Disadvantage : require two memory references to fetch operand

Page No.:
Date: / /

Register Addressing:

Register addressing is similar to direct addressing. The only difference is that the address field refers to a register rather than a main memory address

$$EA = R$$

Advantage

(1) Only a small address field is needed.
(2) no time consuming

Register Indirect addressing:

Indirect addressing is analogous to indirect addressing. only difference is whether the address field refers to memory location or a register

$$EA = (R)$$

Advantage, Disadvantage same as Indirect Addressing

Displacement Addressing:

A very powerful mode of addressing combines the capabilities of direct addressing and register indirect addressing

$$EA = A + (R)$$

Scanned by CamScanner

Instruction type

1. Arithmetic instruction
2. Branch instruction
3. Data transfer instruction
4. logic instruction
5. Bit oriented instruction

Arithmetic used addition, substraction, division, multipl
often execution result stored in first operand

Ex ADD A, Rn -- adds the register to accumulator

SUBB A, Rn => substract register from accumulator with a borrow.

Branch: 10 kind
1. unconditional jump
2. conditional jump

RET => return from subroutine

Data transfer: use for move the content of one register to another

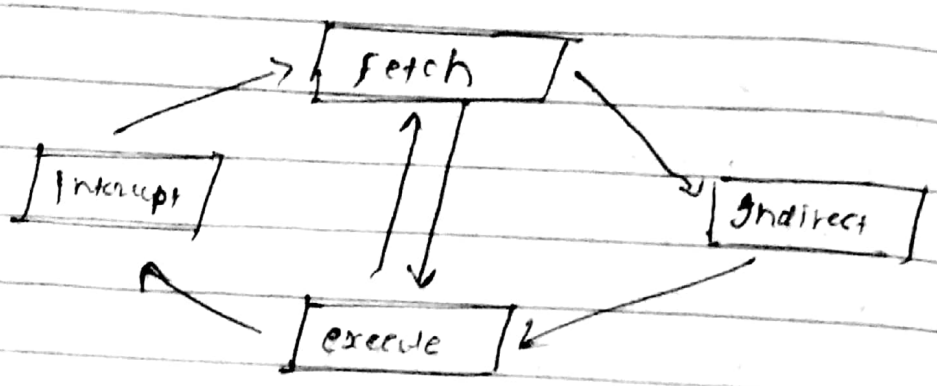
The register the content of which is moved remain unchanged, if they have suffix X (MOVX)
Data core exchanged with external memory

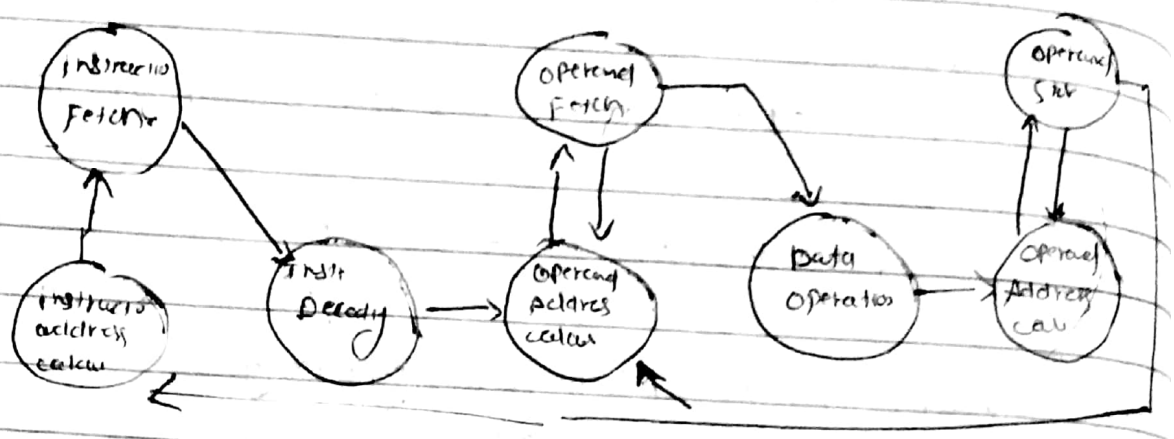
Ex MOV A, Rn

logic instruction used logic operation

ANL A, Rn => AND register to accumulator
ORL A, Rn => OR register to acc

Diagram of Instruction cycle:





instruction cycle State

Use of Program counter.

A program counter is a register in a computer that contain the address of the instruction being executed at the current time. As each instruction get fetch, the program counter increase its store value by 1. After that it point to next instruction.

When computer restart or is reset, P.C normally reverts to 0

Scanned by CamScanner

Access time: (letency)

For random-access memory, this is time it takes to perform a read or write operation.

For non-random access; access time is the time it takes to position the read-write mechanism at desired location

Memory-cycle time: It consist of access time plus any additional time required before a second access can commence

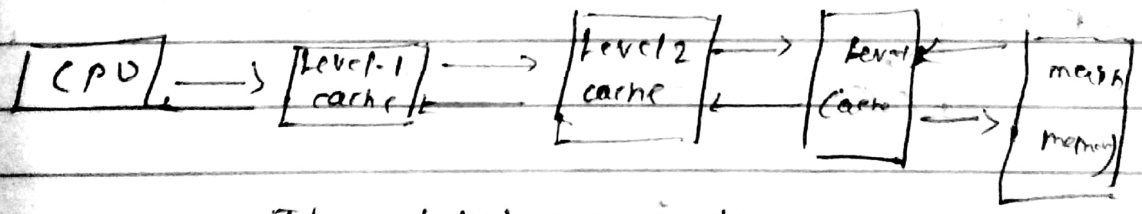
Transfer rate: This is the rate at which data can be transferred into or out of a memory unit.

For random access memory = $\frac{1}{\text{cycle time}}$

Cache memory

Cache memory is fastest memory. at the same time it provide large memory size

The cache contain a copy of portions of main memory

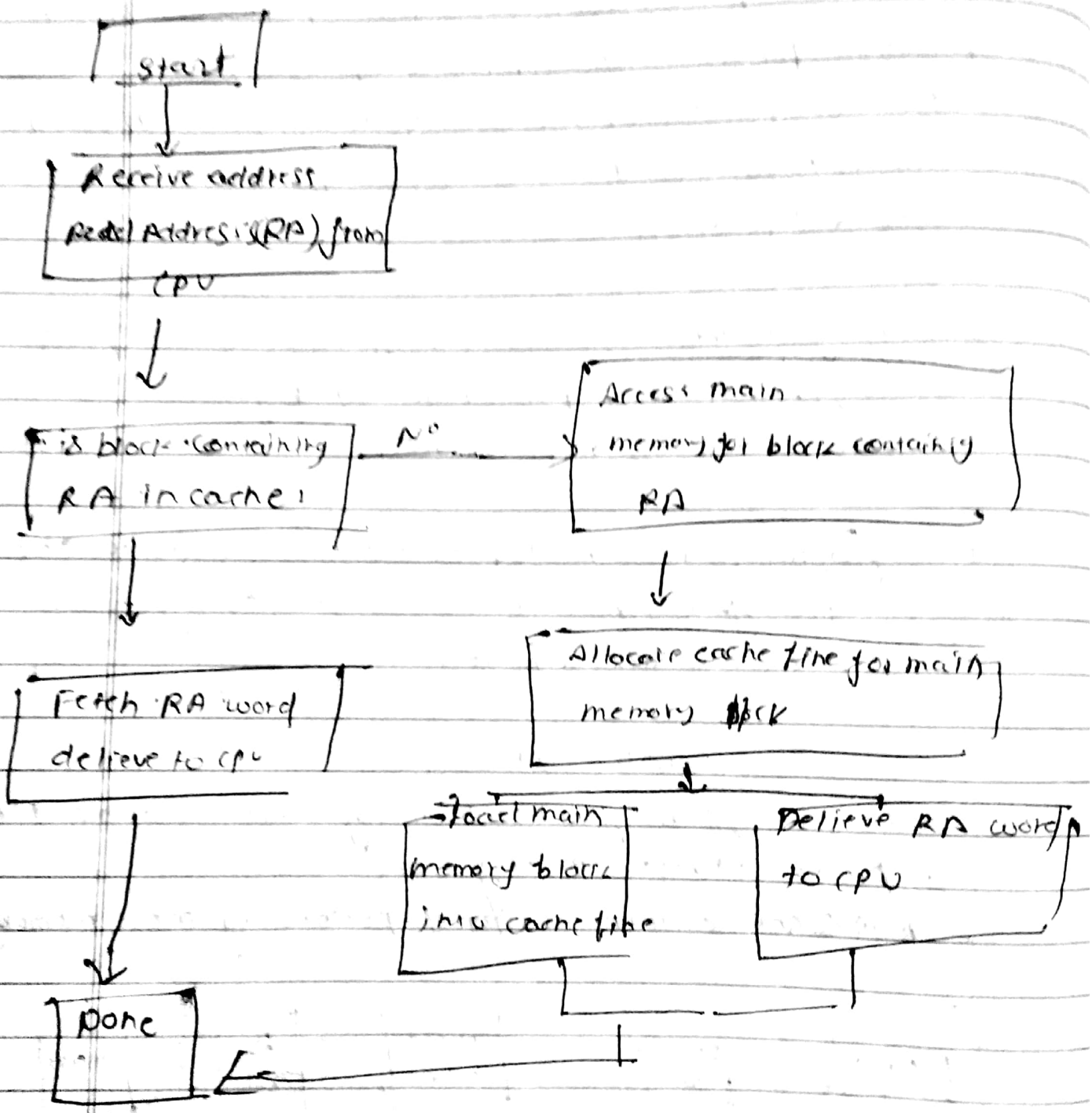


Three-level cache

Scanned by CamScanner

Page No
Date

cache read operation



cache read operation

Scanned by CamScanner

element of cache: -

Cache Address
- logical
- physical

Write Policy
- write through
- write back
- write once

Cache Size

Mapping function
- Direct
- Associative
- Set associative

Number of cache
- single or two level
- unified or split

Replacement Algorithm
- least ~~frequent~~ recently used (LRU)
- First in first out (FIFO)
- least Frequent used (LFU)
- random

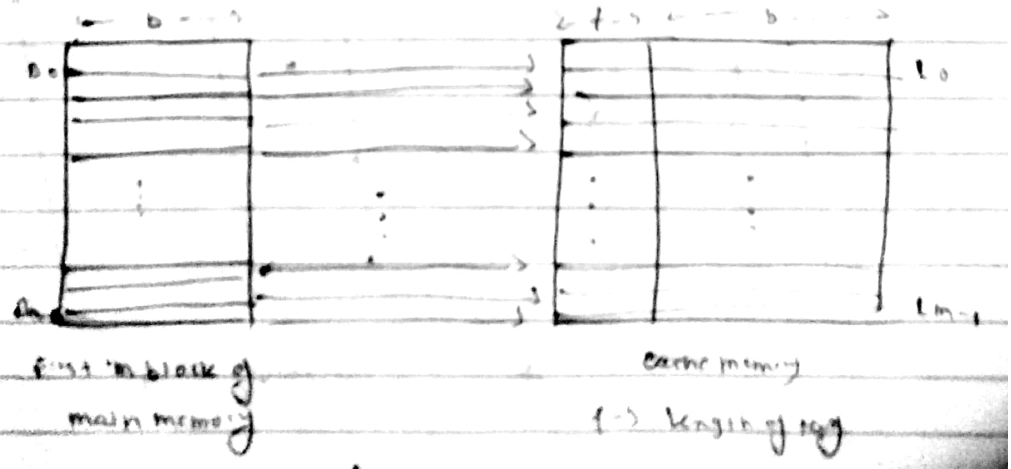
Direct mapping: The simplest technique, map each block of main memory into only one possible cache line.

$$i = j \bmod m$$

i = cache line number

j = main memory block no.

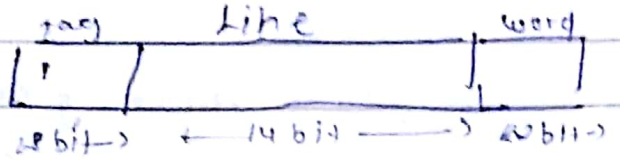
m = no. of line in cache



Direct mapping

Scanned by CamScanner

cache line	main memory block
0	$0, m, 2m, \ldots 2^s - m$
1	$1, m+1, 2m+1, \ldots 2^s - m + 1$
⋮	
$m-1$	$m-1, 2m-1, 3m-1, \ldots 2^s - 1$



Advantage ① simple

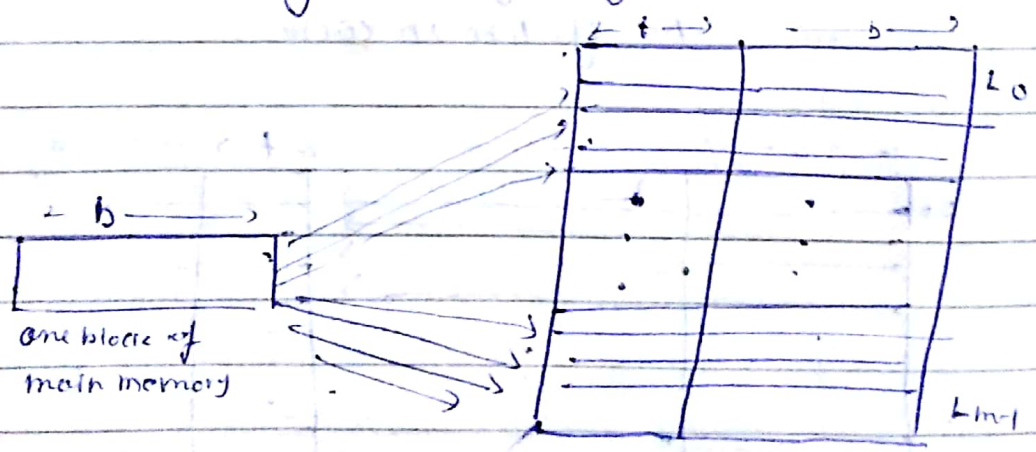
② Inexpensive to implement

Disadv ① 1

① there is a fixed cache location for any given block

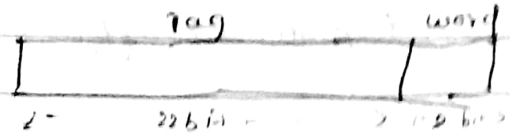
Associative mapping

Associate mapping overcome the discadvantage of direct mapping each main memory block to be loaded into any line of cache In this case, cache control logic interprete a memory address simply as tag and a word feild. The tag field field uniquely identify a block of main memory



Associative mapping

Main memory Address =



Advantage

There is flexibility as to which block to replace when a new block is read into cache

disadvantage associative mapping is the complex circuitry required to examine the tags of all cache line in parallel

Set-Associative mapping:

set associative mapping is a compromise of both Direct and associative approaches Here, cache consists of a number set, each of which consist of a number of lines

$$m = v \times k$$

$$i = j \text{ modulo } v$$

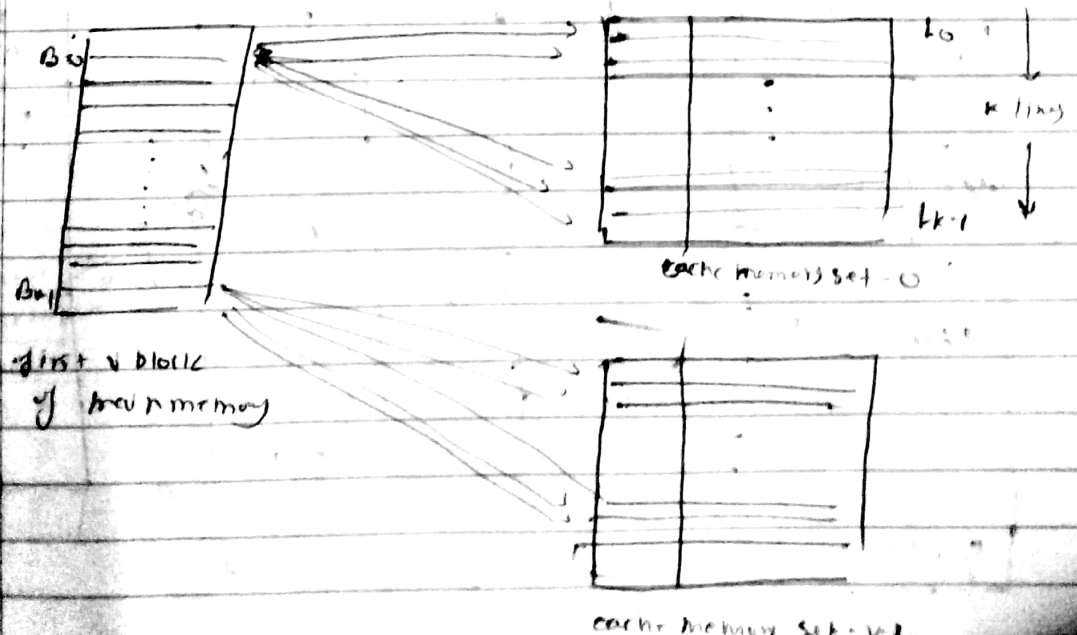
where i = cache set number

j = main memory block

m = no. of line in cache

v = no. of sets

k = no. of lines in each set



Set- A

Replacement algorithm :-

once cache has been filled, when a new block is brough into the cache, one of the existing block must be replaced.

To acheive high speed such algo. needed

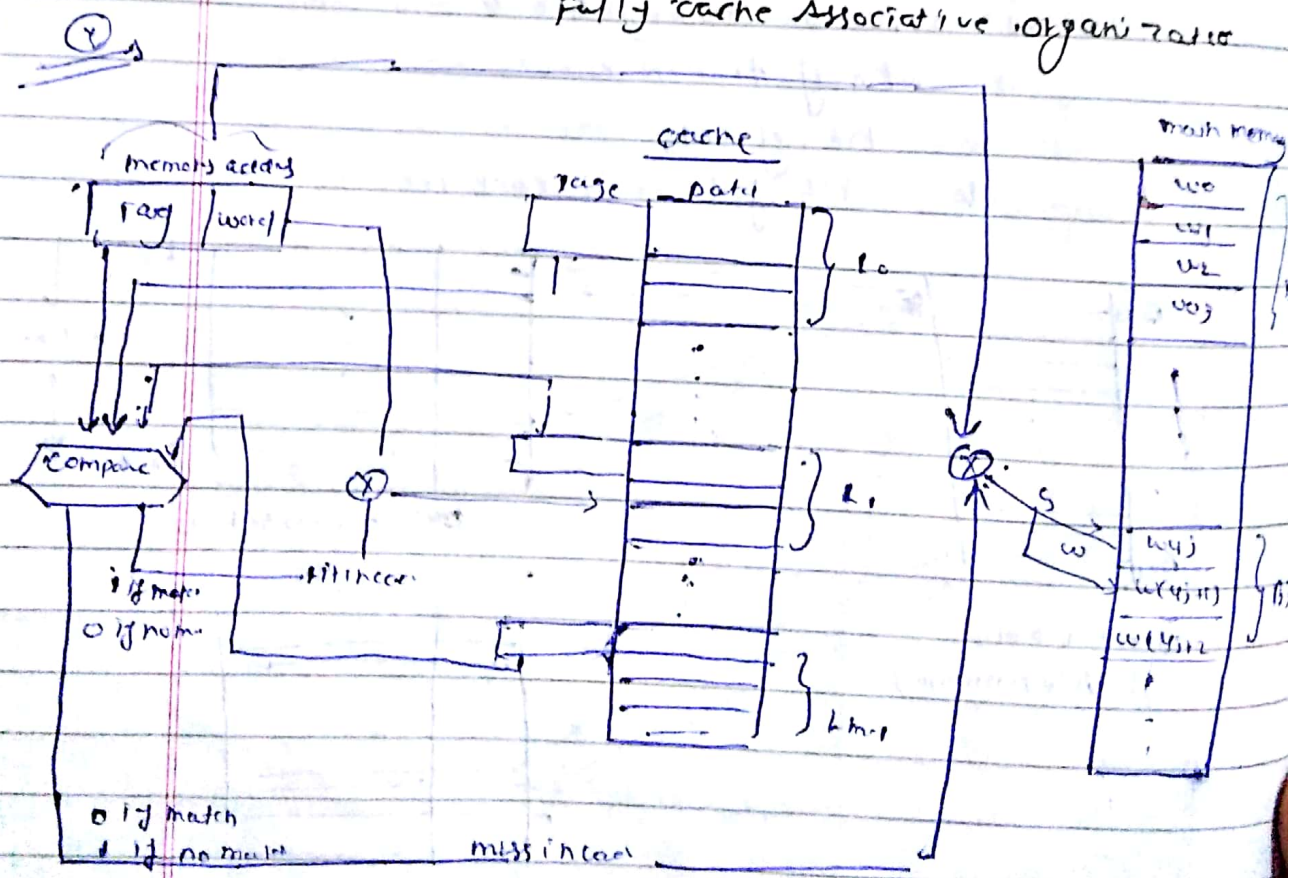
LRU (least recently used) :- Replace that block in the set that has been in the cache longest with no reference to it.

FIFO :- Replace that block in the set that has been in the cache longest.

FIFO is easily implemented as a round robin or circular buffer technique

LFU (Least frequently used) :- Replace that block in the set that has experienced the fewest reference.

Fully cache Associative Organization



Seek time: On the movable head system, the time it takes to position the head at the track is known as seek time. (seek time is the time require to move disk arm to required track)

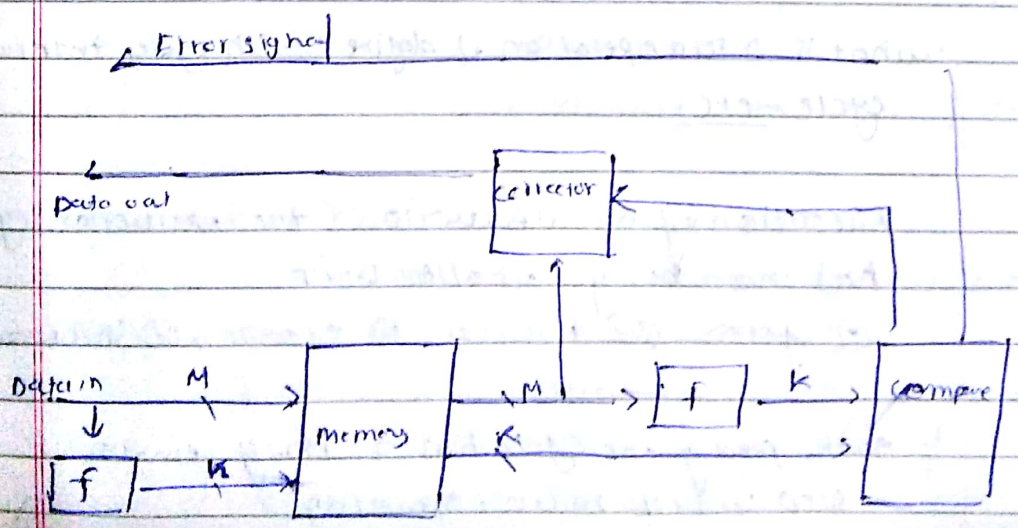
Rotational delay: The time it takes for the beginning of the sector to reach the head is known as rotational delay. or rotational latency

access time: The sum of seek time if any and the rotational delay equal to access time

transfer time: once the head is in proper position the read and write operation is then performed as the sector moves under the head, The time require for the transfer is the transfer time.

(v) Error correction Mechanism in semiconductor memory:

A semiconductor memory system is subject to error. These can be hard failures and soft error.

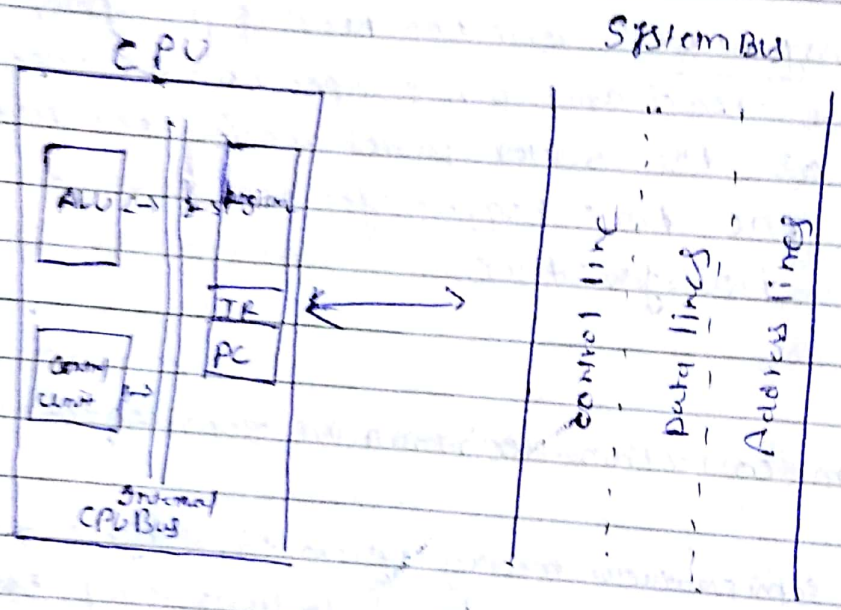


error correcting code function

code that operate in this fashion are referred as error correcting code.

The simplest of the error correcting code is the hamming code devised by Richard Hamming at Bell Laboratories

Internal structure of CPU



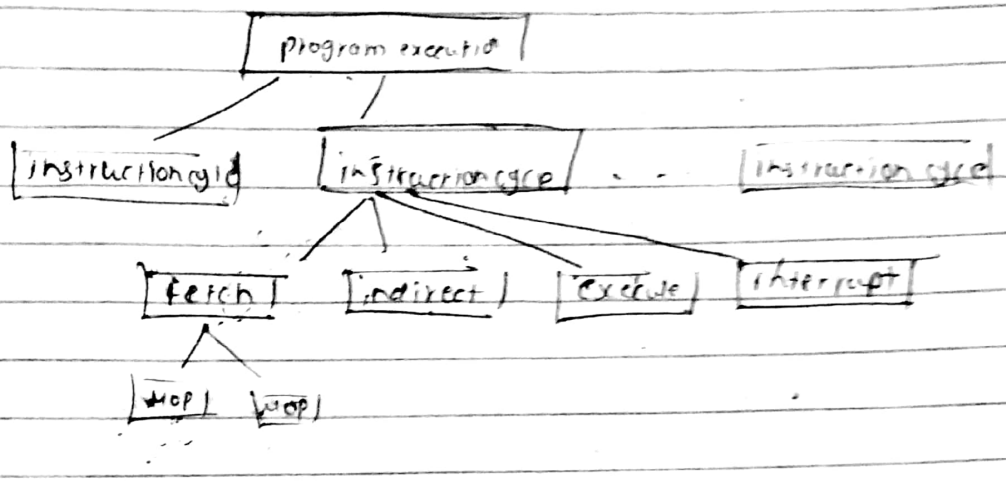
Q what is micro operation. & define fetch cycle interrupt cycle etc

A Execution of an instruction (the instruction cycle) has number of smaller unit
① fetch ② indirect, ③ execute, ④ interrupt

each part of the cycle has a no. of smaller step called micro operation.

Scanned by CamScanner

Micro operation perform basic operation on data stored in one or more registers, including transferring data between registers.



Fetch cycle:

Memory Address register (MAR)
→ connected to address bus
→ specifies address for read write

Memory buffer register (MBR)
→ connected to data bus
→ Hold data

Program counter (PC)
Hold address of next instruction to fetch

Instruction Register (IR)
Hold last instruction fetch

Indirect cycle:

Once instruction has been fetched we need to fetch source operands
→ Assume one address instruction format with direct indirect addressing allowed

t_1 :- MAR ← (IR address)
t_2 - MBR ← (memory)
t_3 → IR address ← (MBR) address

Now IR is updated with Direct address of op

· Interrupt cycle:

At end of execute cycle, processor is interrupt signal. If set, an interrupt cycle occures

t_1: MBR ← (PC)

t_2: MAR ← Save address

t_3: PC ← routine address

t_3: memory ← (MBR)

① This is minimum. Most processor provide multiple type of address.

Difference between Asynchronous and synchronous counter

Asynchronous counter	Synchronous counter
① Clock input is applied to LSB FF. The output of first FF is connected as Clock to next FF.	① Clock input is common to all FF
② All flip-flop are toggle FF	② Any FF can be used
③ Speed depends on no. of FF used for n bit $f_{max} = \frac{1}{n \times t_0}$	③ Speed is independent of no of FF used $f_{max} = \frac{1}{t_p}$
④ No extra logic gates are required cost is less	④ Logic gates are required based on design

Scanned by CamScanner

Scanned by CamScanner

Q list out different control and status register

Register organization

The set of register within the CPU represents the top level of memory hierarchy.

→ ① User visible register: can be accessed by assembly language programmer

② Control and status register: used by control unit to control the operation of the CPU, not directly access by programmer

User visible register: some architecture provide a set of register which can be used without restriction, these are general-purpose register

① Data register: hold only data
② address register: address used

control status register

① Program counter:- hold the address of the instruction to be fetched
② Instruction register (IR):- hold the last instruction fetched
③ Memory Address register (MAR):- hold the address of a memory location that is to be read or written

④ Memory Buffer register (MBR): hold the data to be written to memory most recently

⑤ Program status word (PSW):- condition code

Now IR is updated with direct address of operand

Interrupt cycle:

At end of execute cycle, processor issues interrupt signal. If set, an interrupt cycle occurs

t_1: MBR ← (PC)

t_2: MAR ← Save address

t_3: PC ← routine address

t_3: memory ← (MBR)

- This is minimum. Most processor provide multiple type of address.

Difference between Asynchronous and synchronous counter

Asynchronous counter	Synchronous counter
① Clock input is applied to LSB FF. The output of first FF is connected as clock to next FF.	① Clock input is common to all FF
② All flip-flop are toggle FF	② Any FF can be used
③ Speed depends on no. of FF used for n bit	③ Speed is independent of no. of FF used
$f_{max} = \frac{1}{n \times t_0}$	$f_{max} = \frac{1}{t_p}$
④ No extra logic gates are required	④ Logic gates are required based on design
② cost is less	② cost is more

Scanned by CamScanner

Q. list out different control and status register.

Register organization

The set of register within the CPU represents the top level of memory hierarchy.

⇒ ① User visible register: can be accessed by assembly language programing

② Control and status register: used by control units to control the operation of the CPU, not directly access by programmer

User visible register: some architecture provide a set of register which can be used without restriction, these are general purpose register

① Data register: hold only data
② address register: address used

control status register

① Program counter: hold the address of the instruction to be fetched
② Instruction register (IR): hold the last instruction fetched
③ Memory Address Register (MAR): hold the address of a memory location that is to be read or written

④ Memory Buffer register (MBR): hold the data to be written to memory that recently

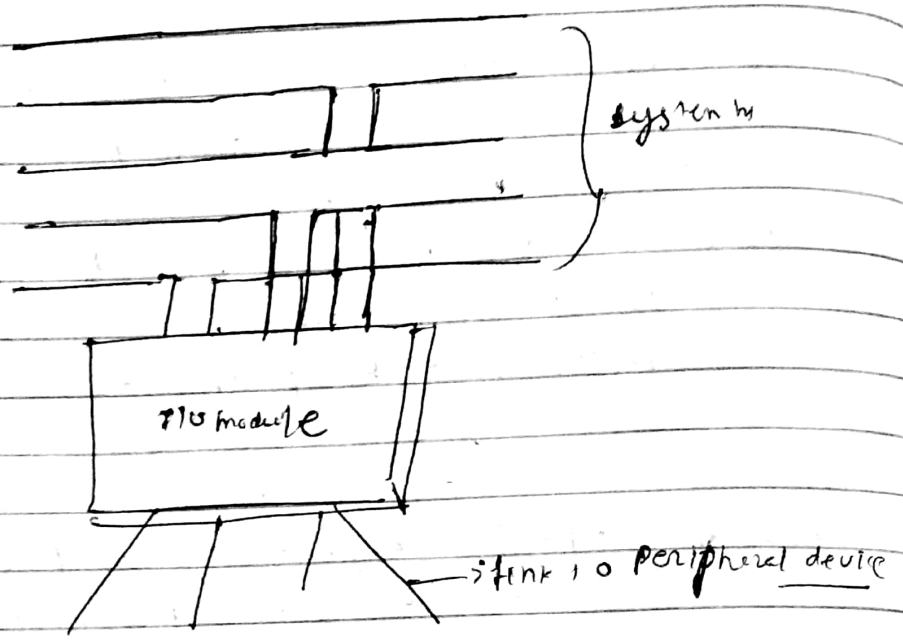
⑤ Program status word (PSW): condition code flag & other bits

Scanned by CamScanner

Page No.
Date / /

Input / output module

The computer system I/O architecture is its interface to the outside world.



I/O module has two major function

① Interface to the processor and memory by system

② Interface to one or more peripheral device by tailored data links

- Delivering different amount of data
- At different speed
- In different formate

Typical I/O steps

① CPU check I/O module device status

② I/O module return status

③ If ready, CPU request data transfer

④ I/O module get data from device

⑤ I/O module transfer data to CPU

Scanned by CamScanner

Types of I/O module:

1. programmed I/O
2. Interrupt - Driven I/O
3. Direct Memory Access

I/O Module function

1. control and timing
2. Processor communication
3. Device communication
4. Data buffering
5. Error detection

1. programmed I/O:

Three technique are possible.

1. Sensing status
2. Read/write commands
3. Transferring Data

Here two circumstance are

1. Memory mapped I/O
2. Isolated I/O

1. Memory mapped I/O:

1. controlling devices just like reading writing memory
2. Requires no special instructions

2. Isolated (special purpose) I/O.

1. No loss of memory address to peripherals
2. simpler address decoding logic
3. can be faster
4. special I/O instruction used

Scanned by CamScanner

Drawback of programmed I/O:

processor has to wait a time for the I/O module of concern to be ready for either reception or transmission of data.

② Interrupt Driven I/O:

* overcome CPU waiting
- No repeated CPU checking of device
- I/O module interrupt when ready

The following sequence hardware event occurs

① Interrupt request signal (device → processor)
② finishes execution of the current instruction
③ Acknowledgement of interrupt
④ Save PC and program state information
⑤ Transfer control to interrupt service routine
⑥ Restore PC and program state information
⑦ Execute next instruction

Design issue: ② problem

① There will almost inveriably be multiple I/O modules, how does processor determine which device issued the interrupt

② If multiple interrupt have occured, how does the processor decide which one to process.?

Scanned by CamScanner

Drawbacks

Any data transfer must traverse a path through the processor.

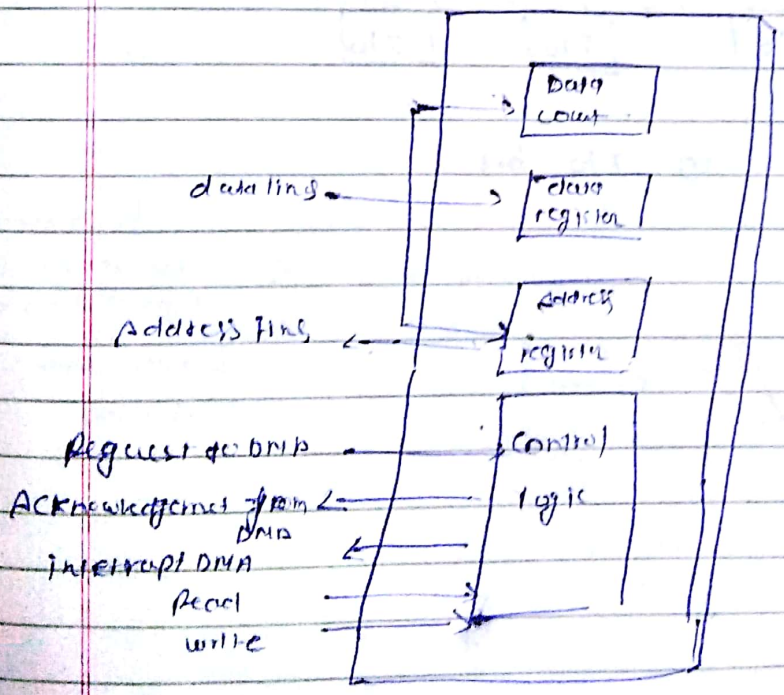
① The I/O transfer rate is limited by the speed which the processor can.

DMA: (Direct Memory Access)

Both interrupt driven and programmed I/O require active CPU intervention

DMA allows devices to communicate directly with memory without passing data through the CPU.

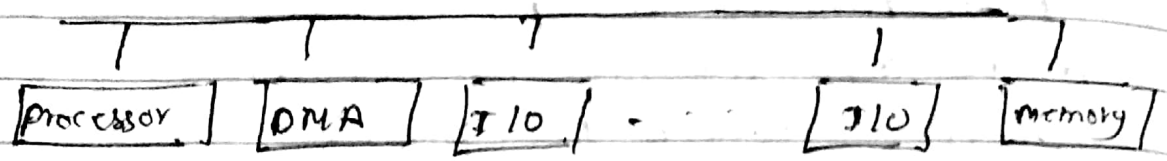
DMA controller ⇒ a device that can control data transfers between I/O subsystem and a memory subsystem in the same manner that a processor can control.



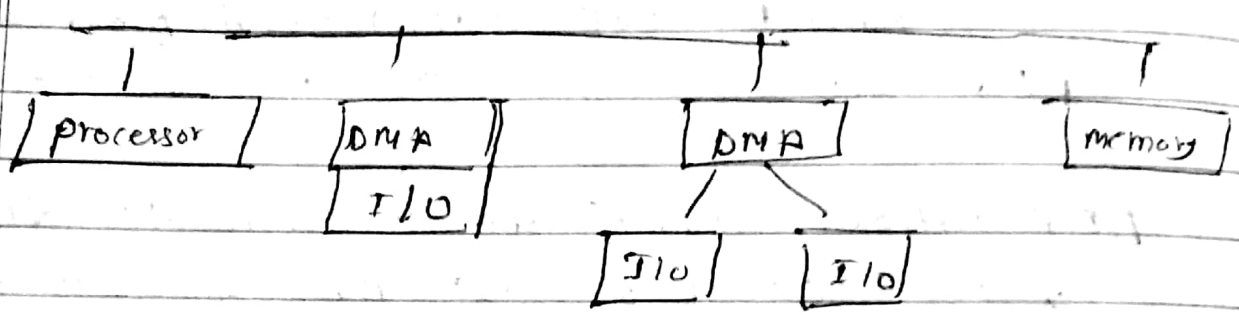
Typical DMA Block diagram

Scanned by CamScanner

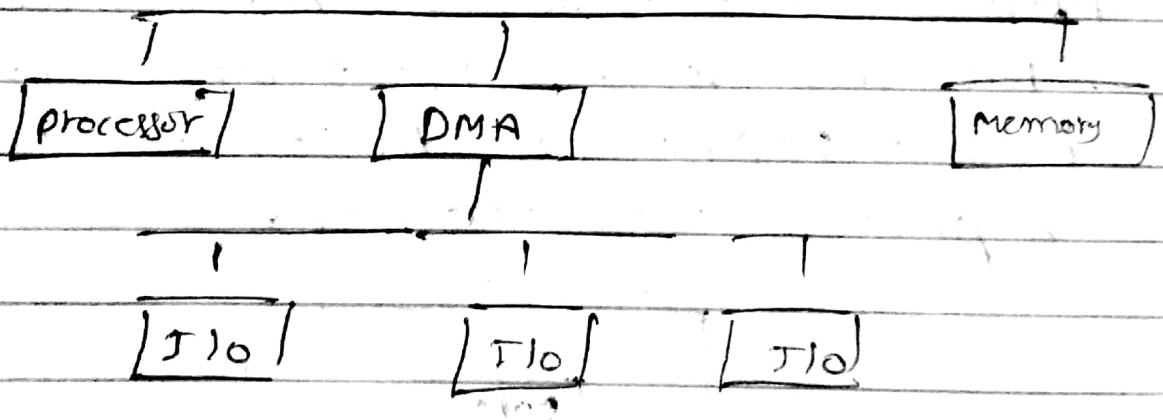
DMA configuration:



(a) single bus detached DMA



(b) single bus, integrated DMA-I/O



(c) I/O bus

Scanned by CamScanner

I/O channels and processors

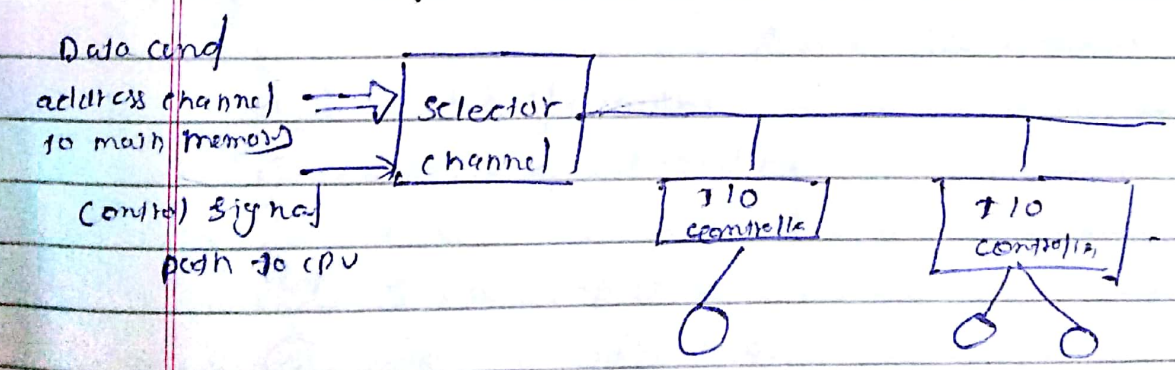
The I/O channel represent an extension of the DMA concept. An I/O chanel has the ability to execute I/O instruction, which gives it compete control over I/O operation

Thus CPU initiates an I/O transfer by instructing the I/O channels to execute a program in memory.

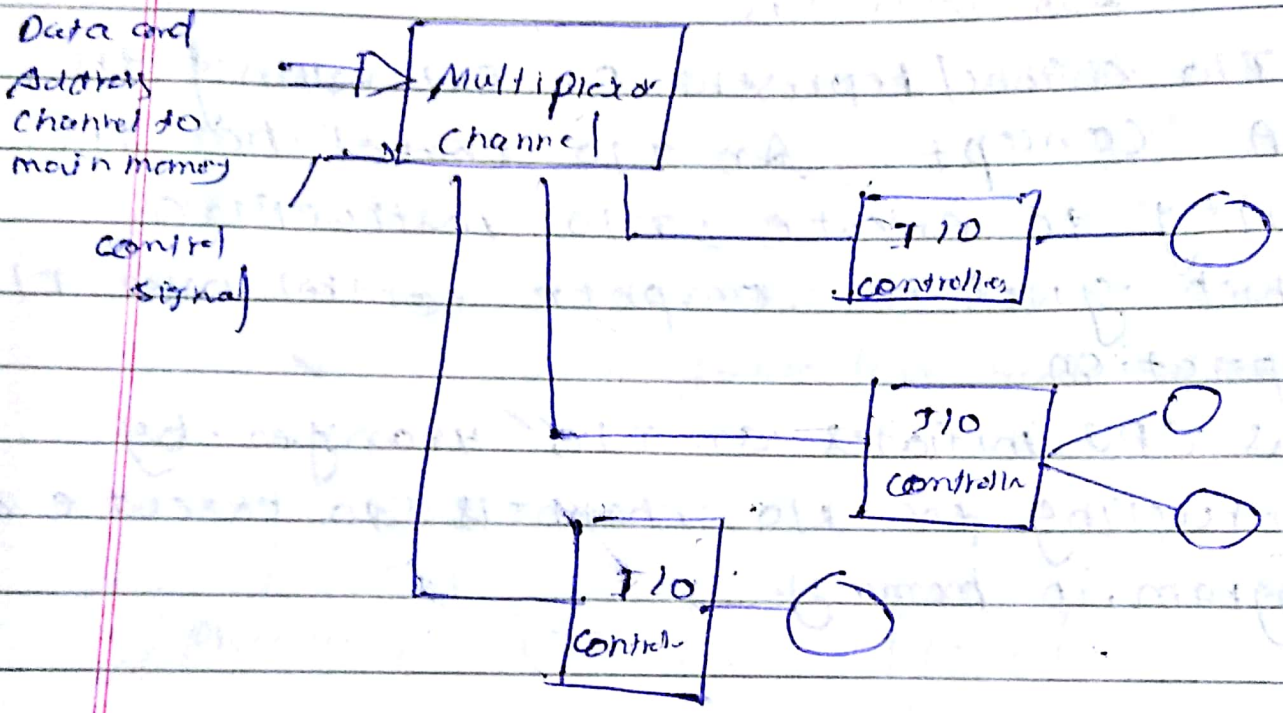
Types of I/O channel

① Selector channel :→ controls multiple high speed device and, at any one time, is dedicated to the transfer of data with one of those devices

② Multiplexor channel :→ can handle I/O with multiple devices at the same time.



(a) selector



(b) Multiplexer.

Scanned by CamScanner

Difference between Direct map, Associative map set associative map

Cache type map	Hit Ratio	Search Speed
Direct mapping	Good	Best
Associative mapped	Best	Moderate
set associative	Very good, Better as N increases	Good, worse as N increases

I/O module function

① control and timing: I/O function include a control and timing requirement, to coordinate the flow of traffic between internal resources and external devices

② processor communication

processor communication involves following

① command decoding
② Data are exchanged
③ Status reporting
④ Address recognition

③ device communication: communication involves commands, status information. etc

Scanned by CamScanner

Page No. :
Date : / /

④ Data buffering:

I/O module must be able to operate both device and memory device.
If the I/O device operate at a rate higher than the memory access rate, then I/O module perform the needed buffering operation

⑤ error detection:-

I/O module is often responsible for error detection
I/O module check the parity to determine whether an error has occured.

Scanned by CamScanner

Page No.
Date: / /

Difference between CISC and RISC

RISC	CISC
(1) Simple instruction taking one cycle	complex instruction taking multiple cycles
(2) It stands as Reduced instruction set computer	(2) It stands complex instruction set computer
(3) Instruction are executed by hardware	(3) Instruction are executed by microprogram
(4) Fixed formate instruction	(4) Variable formate instruction
(5) few instruction	(5) Many instruction
(6) Multiple register set	Single register set
(7) Highly pipelined	~~Single~~ No pipelined or less pipelined
(8)	

Scanned by CamScanner

pipelling:

pipeling is an implementation technique whereby multiple instruction are overlapped in execution.
This is solved without additional hardware but only by letting different parts of the hardware work for different instruction at the same time

Six stage of pipeline:-

1. FI → fetch instructions , ~~FO → fet~~
2. DI → decode instruction
3. CO, calculate operand address
4. FO → fetch operand
5. EI → execute instruction
6. WO → write operand

FI	DI	CO	FO	EI	WO		
	FI	DI	CO	FO	EI	WO	
		FI	DI	CO	FO	EI	WO
			FI	DI	CO	FO	EI	WO

Fetch	execute	Memory	Store	
	fetch	execute	memory	Store
		fetch	execute	memory	Store

pipeling

TNPL

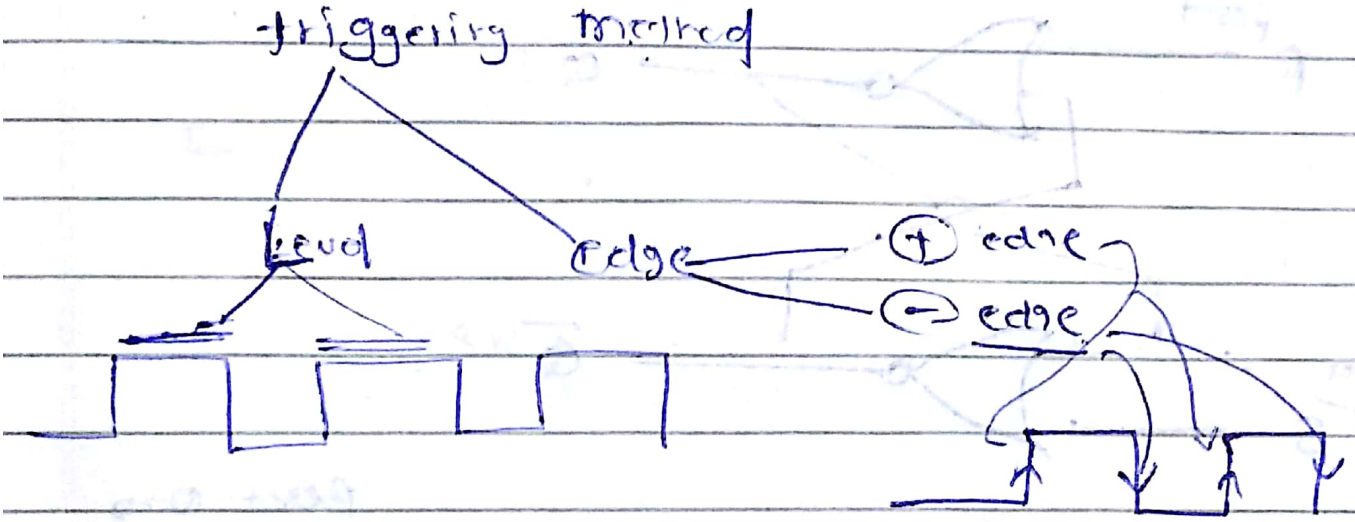
Page No. :

Date : / /

Clock

Duty cycle :- $\frac{\text{Signal}\uparrow}{\text{Total time}}$

Triggering method



Difference between latch | flipflop

~~Sf~~ - Latch is ~~ed~~ level trigger

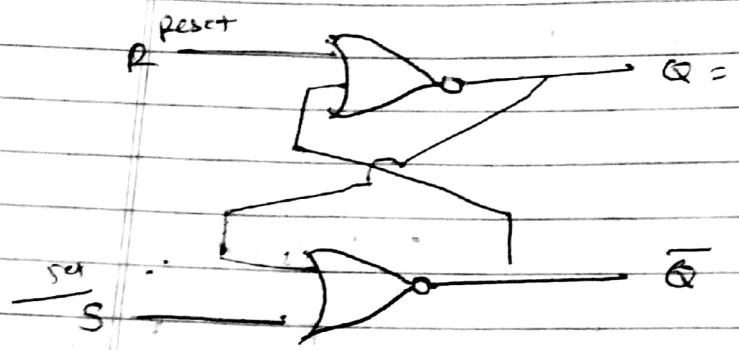
flipflop is edge trigger

Scanned by CamScanner

SR Latch

- The basic storage element is called latch.

using nor gate



Reset Q=0
set Q=1

case I = S=0, R=1, Q=0, $\bar{Q}$=1

S=0, R=0, Q=0, $\bar{Q}$=1 memory

case II. S=1, R=0, Q=1, $\bar{Q}$=0
S=0, R=0, Q=1, $\bar{Q}$=0 memory

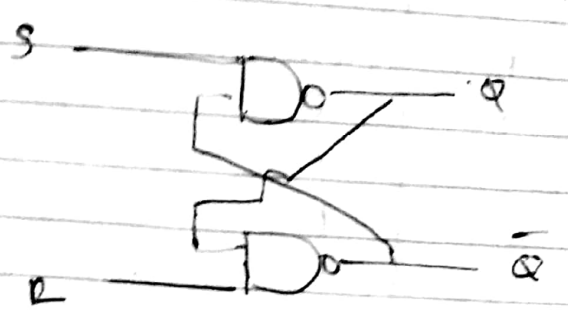
Truth

case III ⇒ S=1, R=1 Q=0 $\bar{Q}$=0
S=0, R=0 Q=0 $\bar{Q}$=1

S	R	Q	$\bar{Q}$
0	0	memory as before	
0	1	0	1
1	0	1	0
1	1	forbidden state (Not used)	

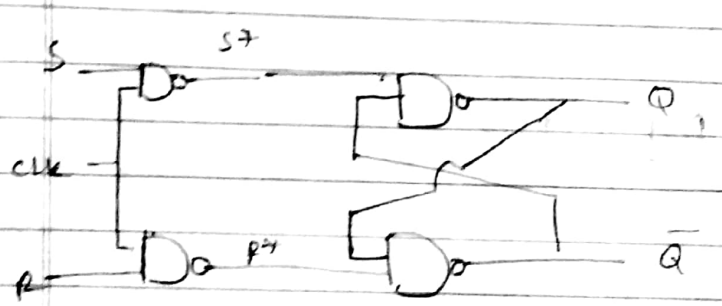
Scanned by CamScanner

Using NANDgate



S	R	Q	$\bar{Q}$
0	0	not used	
0	1	1	0
1	0	0	1
1	1	memory	

SR flipflop



$S^* = (\overline{S.CLK}) = \bar{S} + \overline{CLK}$

$R^* = (\overline{R.CLK}) = \bar{R} + \overline{CLK}$

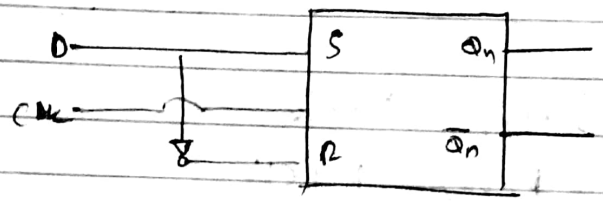
Truth table

CLK	S	R	Q	$\bar{Q}$	Next state Q_{n+1}
0	X	X	memory		Q_n
1	0	0	memory		Q_n
1	0	1	0	1	0
1	1	0	1	0	1
1	1	1	Not used case		invalid

Characteristic table

Q_n	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

D flip-flop



Truth table

Clk	D	Q_{n+1}
0	X	Q_n
1	0	0
1	1	1

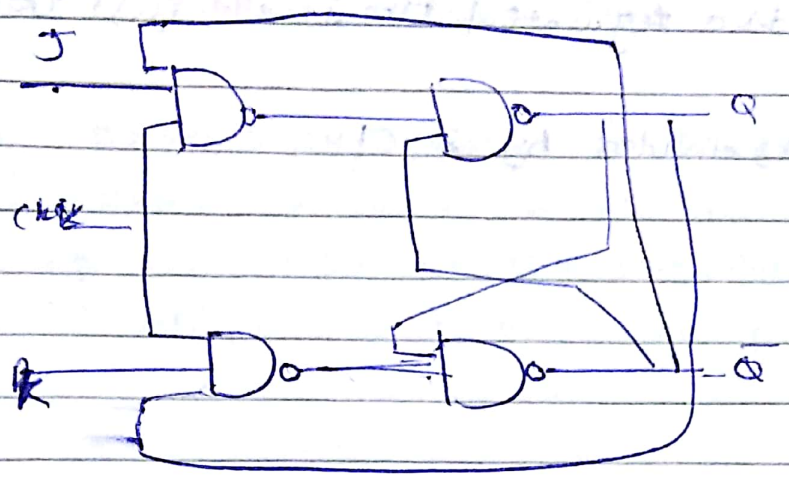
Charcl tcn

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

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

Scanned by CamScanner

JK flip-flop



Truth table

Clk	J	K	Q_{n+1}
0	X	X	Q_n } memory
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	$\overline{Q_n}$ toggle

Char

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

Scanned by CamScanner

Page No.:
Date: / /

Counter

counter is a sequential circuit. It just count.

FF as divided by 2 CLK:

Scanned by CamScanner

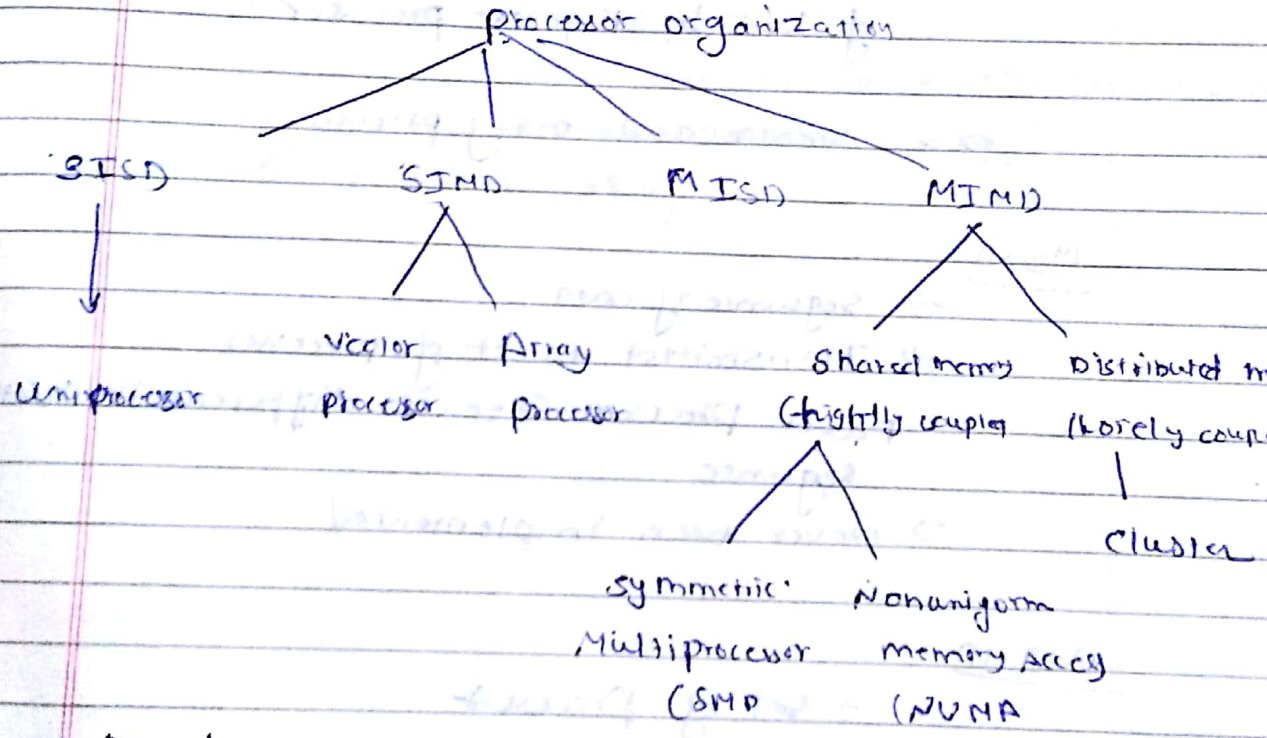
write short note on (parallel processing)

processor Architecture

Multiple processor organization

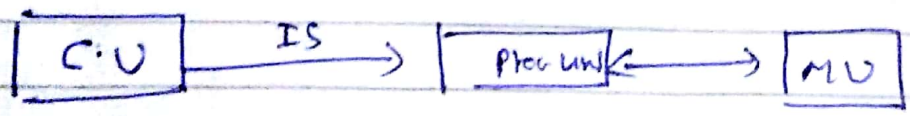
1. SISD (single instruction, single data stream)
2. SIMD, (single instruction, multiple data stre)
3. MISD (multiple instruction single data stream)
4. MIMD (multiple instruction, multiple data stream)

Taxonomy of parallel processor Architecture



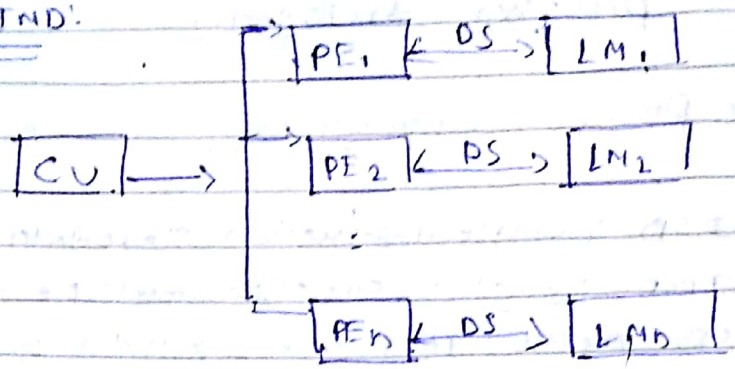
SISD:-

Single processor, single instruction stream
Data stored in single memory
Uni-processor



Scanned by CamScanner

SIMD:



→ Single machine instruction
→ control simultaneous execution
→ each instruction executed on different set of data by different processor

Ex → Vector and array processor

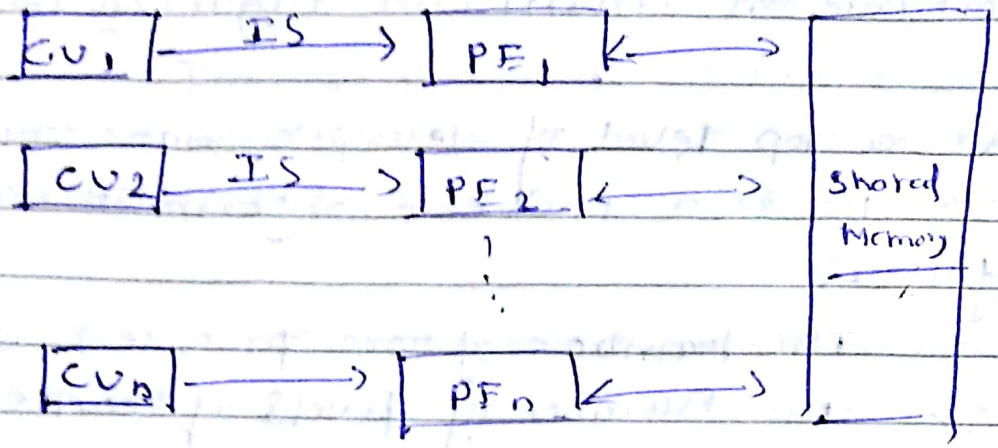
MISD:-

→ Sequence of data
→ Transmitted to set of processor
→ Each processor executes different instruction sequence
→ Never been implemented

MIMD:-

→ set of processor
→ simultaneously execute different instruction sequences
→ different sets of data
Ex: SMP, Cluster and Numa systems

Scanned by CamScanner



Challenge in parallel processing

1. communication cost
2. synchronization cost
3. Not all problem are amenable to parallelization
4. Hard to think in parallel
5. Hard to debug

Scanned by CamScanner

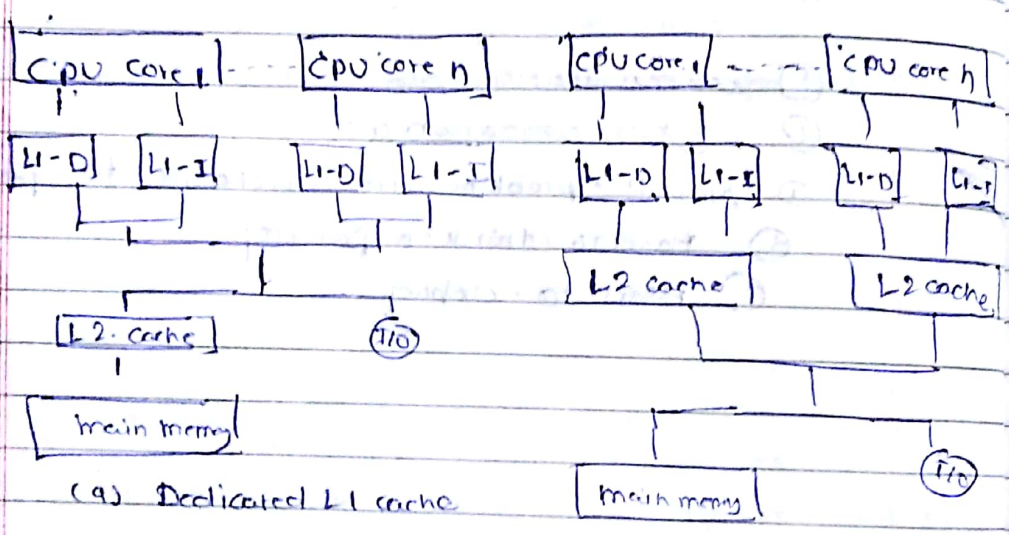
Def Multicore Processor: A multi core processor is one with combining two or more independent processor into a single package, often a single integrated circuit

Page No.
Date: / /

short note on multi core organization

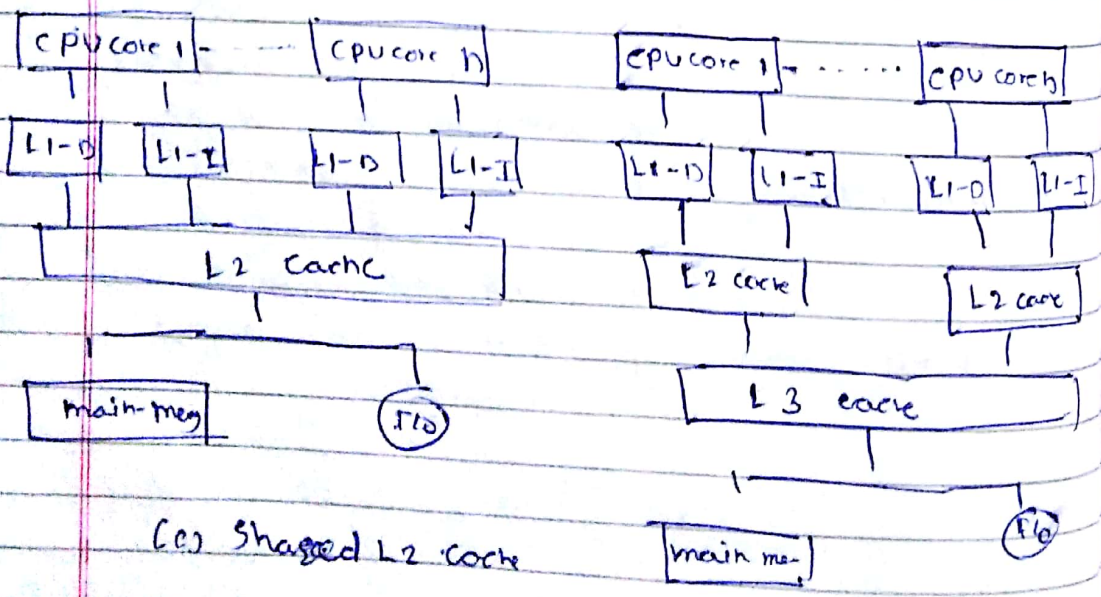
At a top level of description, the main variables in a multicore organization are as follows

- → The number of core processors on the chip
- → The number of levels of cache memory
- → The amount of cache memory that is shared



(a) Dedicated L1 cache

(b) Dedicated L2 cache



(c) Shared L2 cache

(d) Shared L3 cache

Scanned by CamScanner

Page No.
Date / /

Advantage

The use of shared L2 cache on the chip has several advantages over exclusive reliance on dedicated caches

1) · Constructive interference can reduce overall miss rate

2) A related advantage is that data shared by cores is not replicated at the shared cache level

3) · Interprocessor communication is easy to implement via shared memory location

Different architecture

① Intel X86 multicore organization
② Intel core i7

D-

Advantage of multi core processor

① provide more pc security
② It will work faster for certain program
③ The signal between different CPU travel shorter distance

Disadvantage

① They get only 60-30% more speed
② Their cost more than single core processor
③ difficult to manage

Scanned by CamScanner

Cloud computing

The concept of cloud computing evolved in 1's 10 th 2006 Amazon provided first public cloud AWS (Amazon web services)

Cloud computing

The term cloud refers to a network or internet. We can say that cloud is something, which is present at remote location. Cloud can provide services over network.

Ex. Gmail, hotmail run in cloud.

Types of cloud

1. public cloud: → open for public
   Ex → E-mail

2. Private cloud: used by organization ie isolated offer security.
   Ex → Banks

3. community cloud: access by group of organizations
   Ex. All government agencies

4. Hybrid cloud:
   mixture of public and private

Scanned by CamScanner

Service model of Cloud:

There are three basic Service model which are

1. Software as a Service (SaaS)
2. Platform as a Service (PaaS)
3. Infrastructure as a Service (IaaS)

1. SaaS:

Describe any cloud service where consumer are able to access software application over the internet.

ex: Google, Twitter, Facebook etc

Benefits:
- No additional hardware cost
- No initial set up
- easy to use
- Automatic update
- Access from any location

2. PaaS:

Allows user to create Software application using tools supplied by the provider. It is accessed by user via web browser

ex: Web dev., App dev., Programming language execution environment

Benefits:
1. Flexibility
2. Security
3. Makes development possible for non-exp

Infrastructure as a Service

It provides access to fundamental resource such as physical machine, virtual machine, virtual storage etc

Ex: · Amazon web services (AWS)
google compute Engine.

Benefits

Enterprise infrastructure
No investment in hardware
Utility style costing

Cloud com

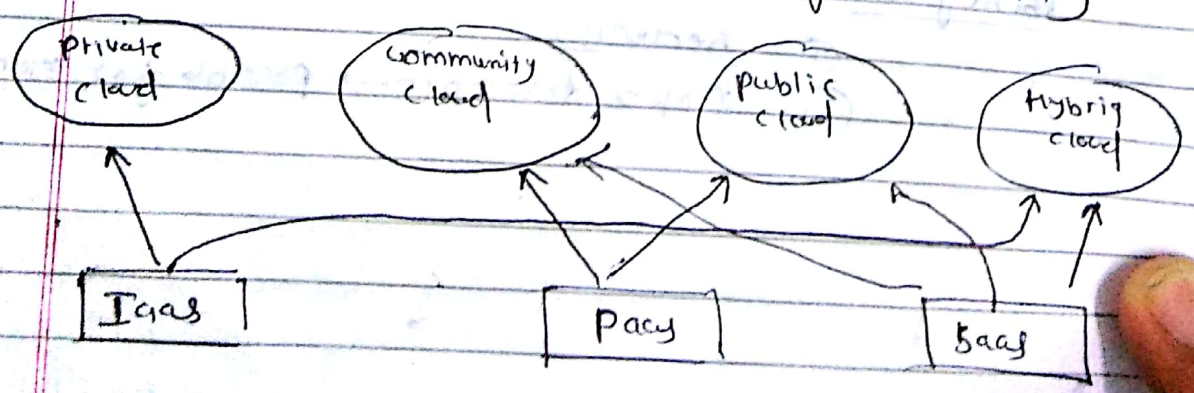
Advantage

1. Access your data at all time
2. Storage centre is no longer
3. Payment when used
4. easily scalable

Disadvantage

1. Less control over data
2. Dependence on a third party
3. Long term Dependence

Imp Deployment model of cloud computing

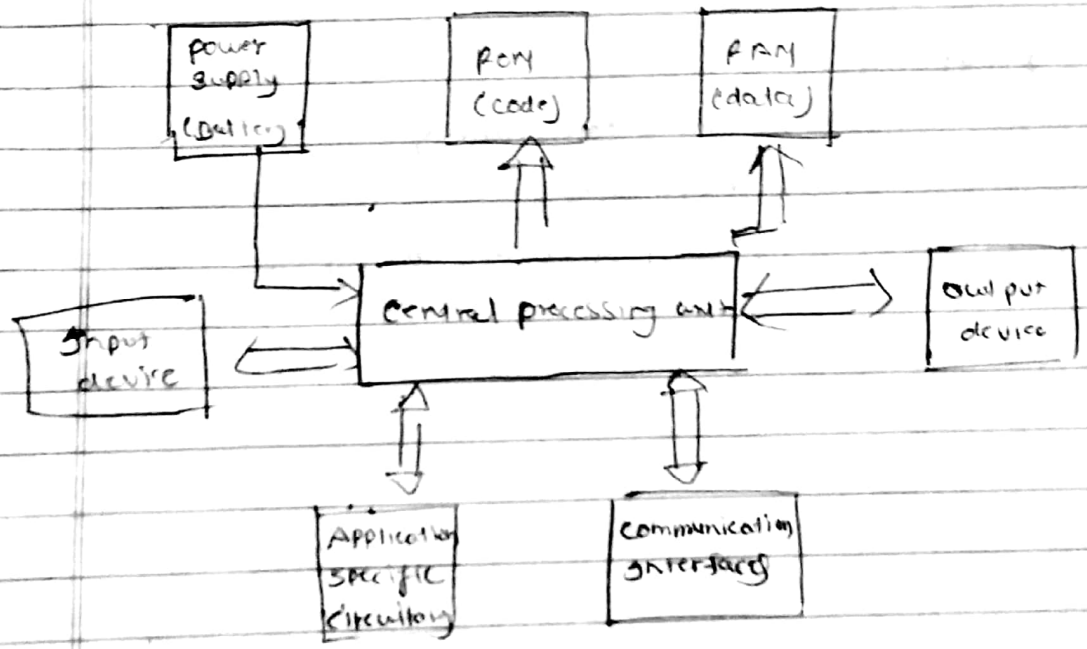


Scanned by CamScanner

Imbedded System

A combination of computer hardware and software, and perheps additional mechanical or other parts, designed to perform a dedicated function.

Block Diagram



① CPU · ALU, CU.

② Memory · Embedded system memory can be either on chip or off chip on-chip memory access is much fast than off-chip memory, but size of on-chip is much smaller

③ I/O ports keypads, switches, buttons, sensor.

Scanned by CamScanner

Mobile embedded System

Mobile embedded system used in portable embedded device like cell phones, mobiles, digital camera, PDA (personal digital assistance).

The basic limitation of these device is the power resource and limitation of memory

Limitation

- memory constraints
- Small size
- Lack of good user interface

Scanned by CamScanner

Advantages of RISC

1. RISC is more concerned with software, and favors single clock, reduced instruction as opposed to complex
2. Speed of the operation can be maximized and the execution time can be minimized
3. Simpler and more easy to use hardware

Disadvantage

1. Mostly the performance of the RISC processor depends on the programmers or compiler. Because compiler change CISC code to RISC code
2. The first level cache of RISC processor is also a disadvantage of RISC

Advantage of CISC:

1. Microprogramming is easy assembly language to implement and less expensive than hard wiring a control unit
2. The compiler has to do very little work to translate a high-level language statement into assembly
3. Very little RAM is required to store instruction

Disadvantage

1. The performance of the machine slow down due to the amount of clock time taken by different instruction will be dissimilar

Scanned by CamScanner

RAID (Redundant Array of Independent Disk)

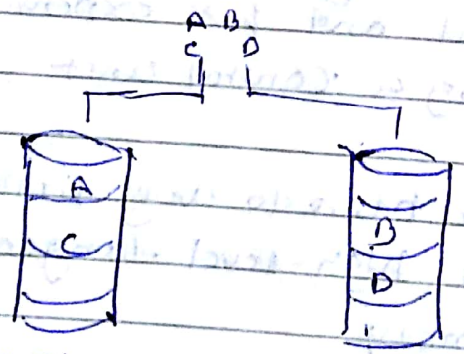
RAID is a storage technology that combines multiple disk drive component into a logical unit

Raid levels
(i) Raid level 0
Raid level 1
Raid level 2
Raid level 3
Raid level 4
Raid level 5

Raid level 0:

Raid level 0 divides data into block unit and write them across number of disks

As data is placed across multiple disk; it is called data stripping



Advantage:

Increase speed
implementation easy
no overhead of parity calculation

disadv:

The failure of drive will result in all data being lost

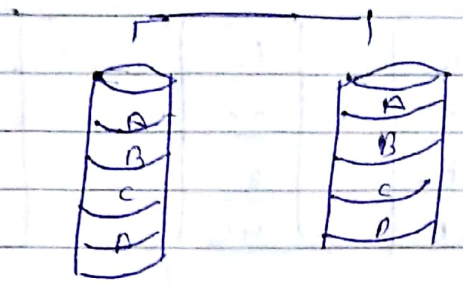
Scanned by CamScanner

Page No.:
Date: / /

Raid level 1:

This level is called "mirroring" as it copies data onto two disk drive simultaneously

Data can be read from either disk but it written on both disk



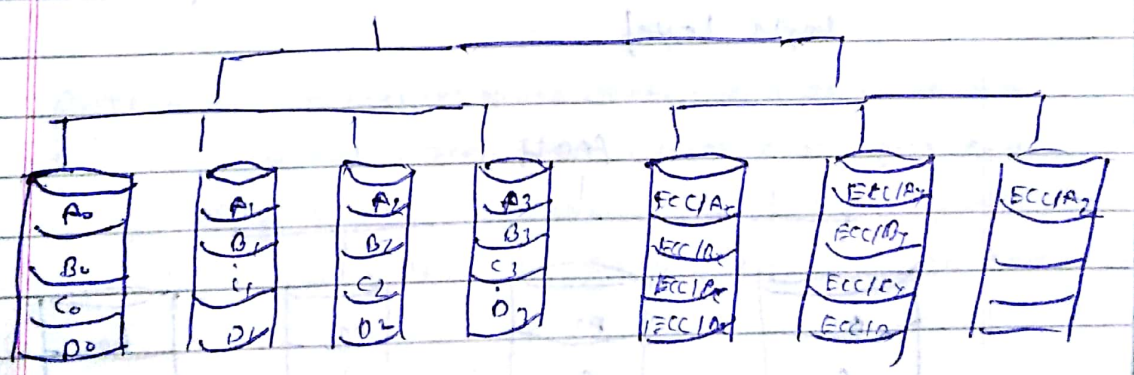
Advantage ① If one disk fails, other automatically takes over.

② raid 1 is used to store system software.

Disadvantage

cost, storage cost increase

Raid level 2: (Bit level striping with dedicate hamming code parity).



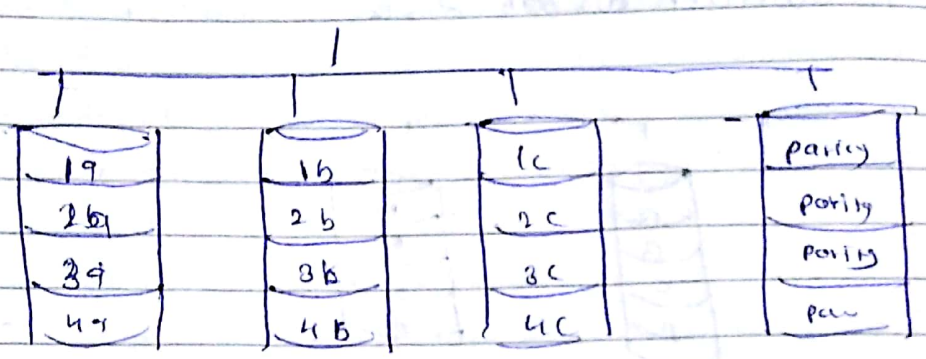
Advantage

① Data error correction

② Extremly high data transfer rates parity

RAID level 3 (Byte level striping with a dedicated parity disk)

① written in parallel on two or more device
② An additional drive store parity information

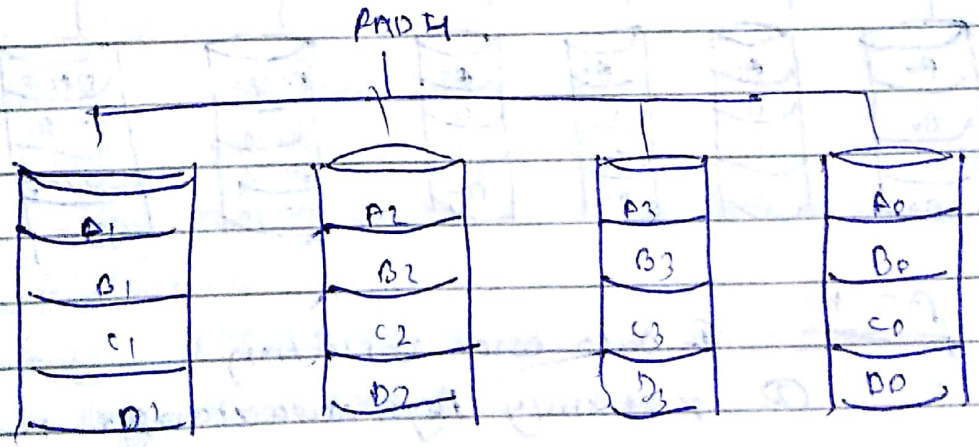


Advantage: ① R/W data transfer is very high
② Disk failure has a minor impact on data

Disad: ① If the parity disk fails, you'll end up RAID 0

RAID 4 (Block-level striping with dedicated parity disk)

Raid 4 is similar to RAID 3 except at the point it works at a block level and not byte level



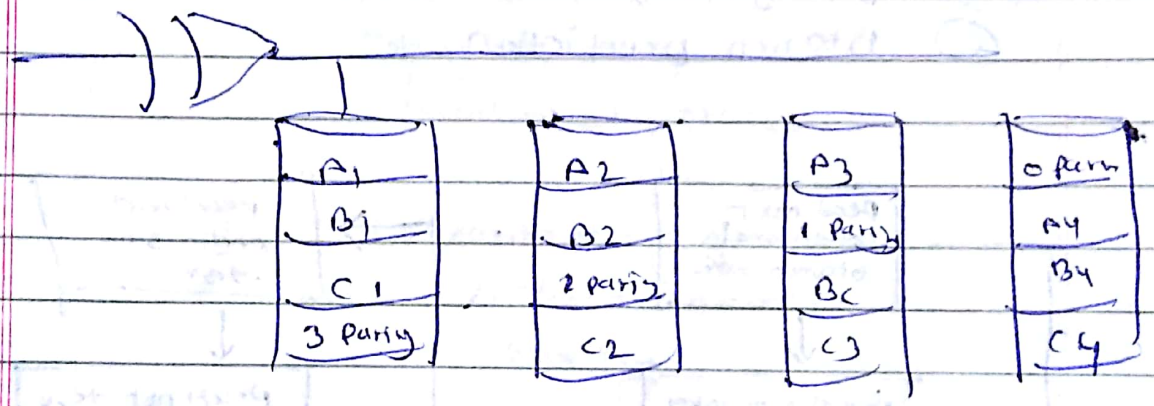
Scanned by CamScanner

Page No.
Date / /

Advant ① Reading operation is very fast
② fault tolerance is very high

Raid 5: (Block level striping with parity data distribution across all disk)

=> RAID 5 is similar to RAID 3 except that the parity is distributed to all drives



Advan ① Read data transaction are very fast while write data are somewhat slow

Disad: This is complex technology

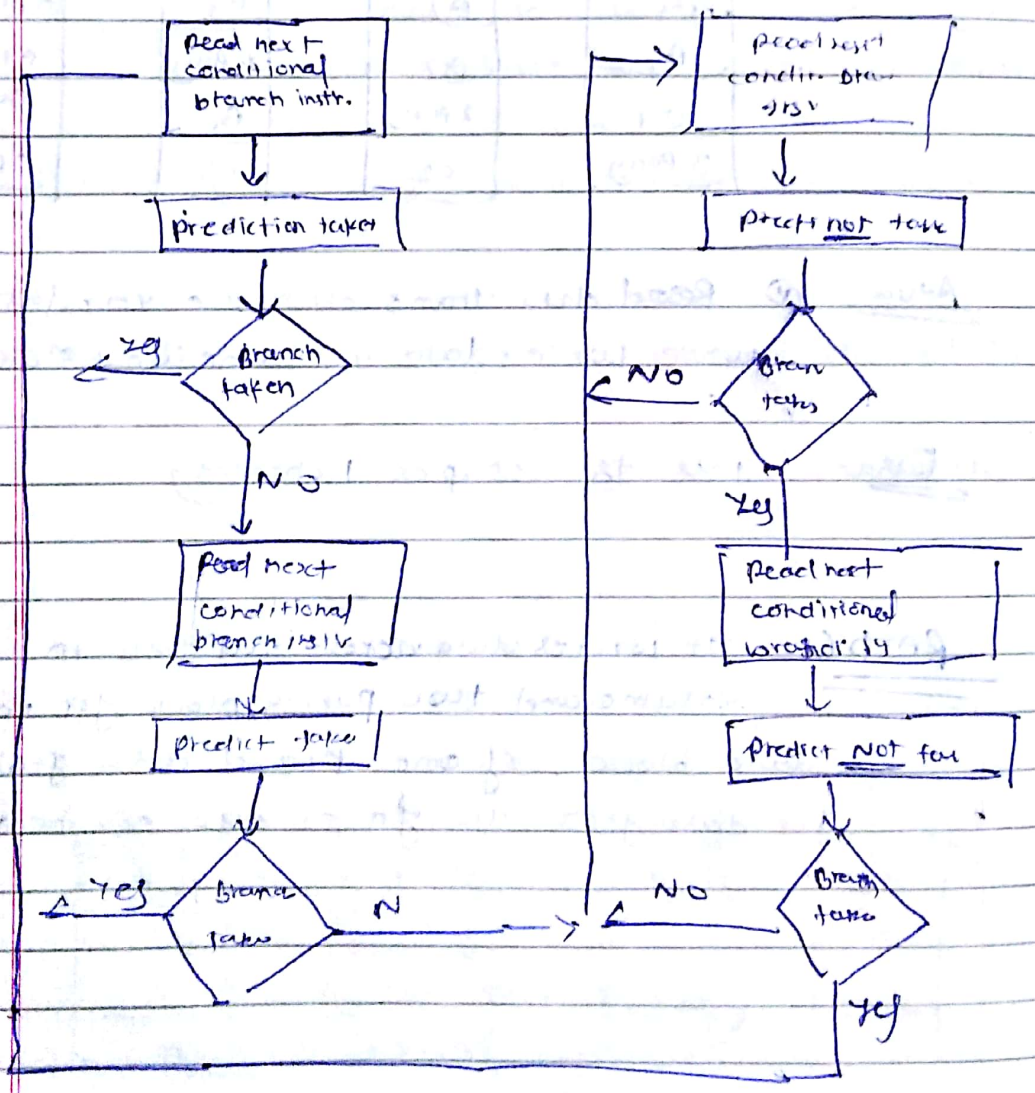
RAID 6: It writes data across all disk in the volume and two parity block for each data block. If one physical disk fails, the data from the failed disk can be rebuilt

Scanned by CamScanner

Instruction pipelining

One of the major problem in designing an instruction pipeline is conditional branch instruction. A variety of approaches have been taken for dealing with conditional branches

1. Multiple streams
2. prefetch branch target
3. Loop Buffer
4. Delayed Branch
5. Branch prediction



Branch Prediction flowchart

Scanned by CamScanner

Virtual memory

To run a large program. (1961)

Terms in virtual

1. Address space :->
2. Page
3. frame
4. maping

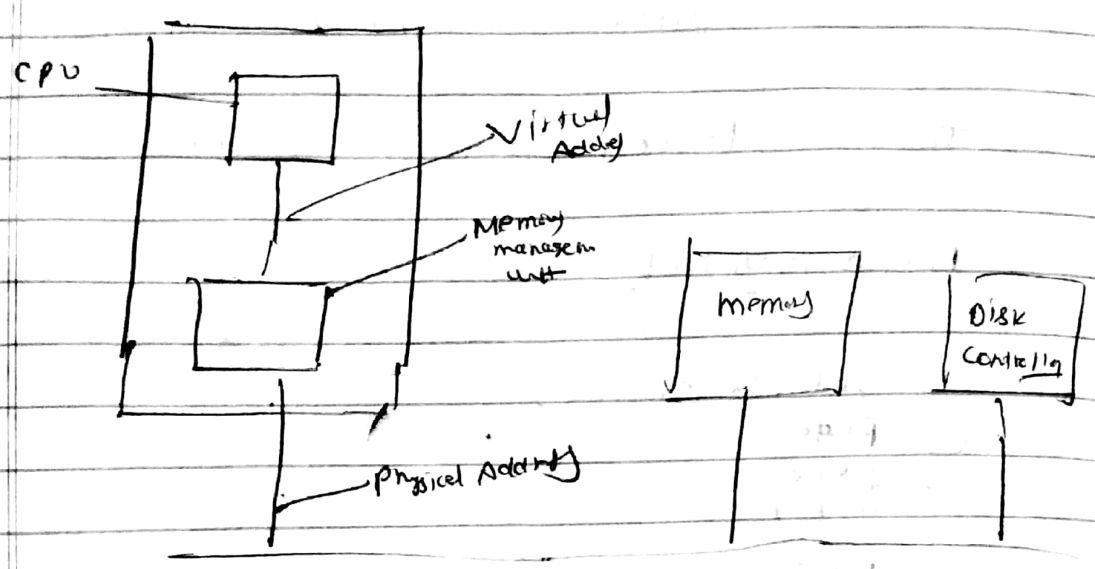
page :-> continuous range of address
logical concept
equal size
present in any order

Frames :-> counterpart of page
physical concept
reality
equal size

Address space :-> Range of address
owned by each program
Broken into pages

:-> Mapping:
mapping between page and frames
Heart of virtual memory
called Memory managem unit

Scanned by CamScanner



simple paging

formal

$L.A = \log_2(L.A.S)$

$P.A = \log_2(PAS)$

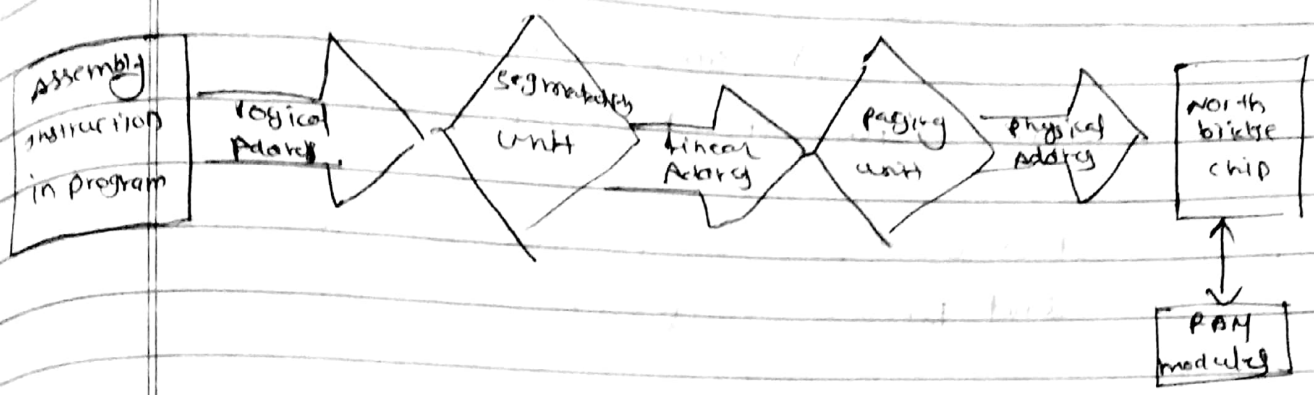
No. of pages $= LAS / P.S$

paging: Memory divided into equal-sized frames. All process pages loaded into non-necessarily contiguous frames

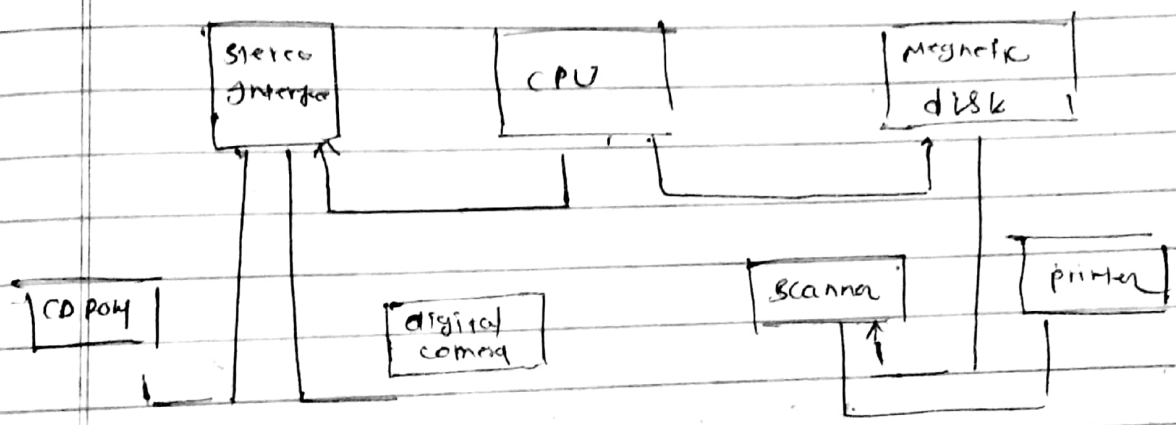
segmentation: each process divided into variable size segments. All process segments loaded into dynamic partition that are not necessarily contiguous

A program is a collection of segments

Scanned by CamScanner



Simple Firewire configuration



Firewire =>

Firewire is a method of transfering information between digital device, especially audio and vedio equipment

Type of interface

1. User interface
2. Hardware interface
3. System interface

Scanned by CamScanner

Infini Band :-

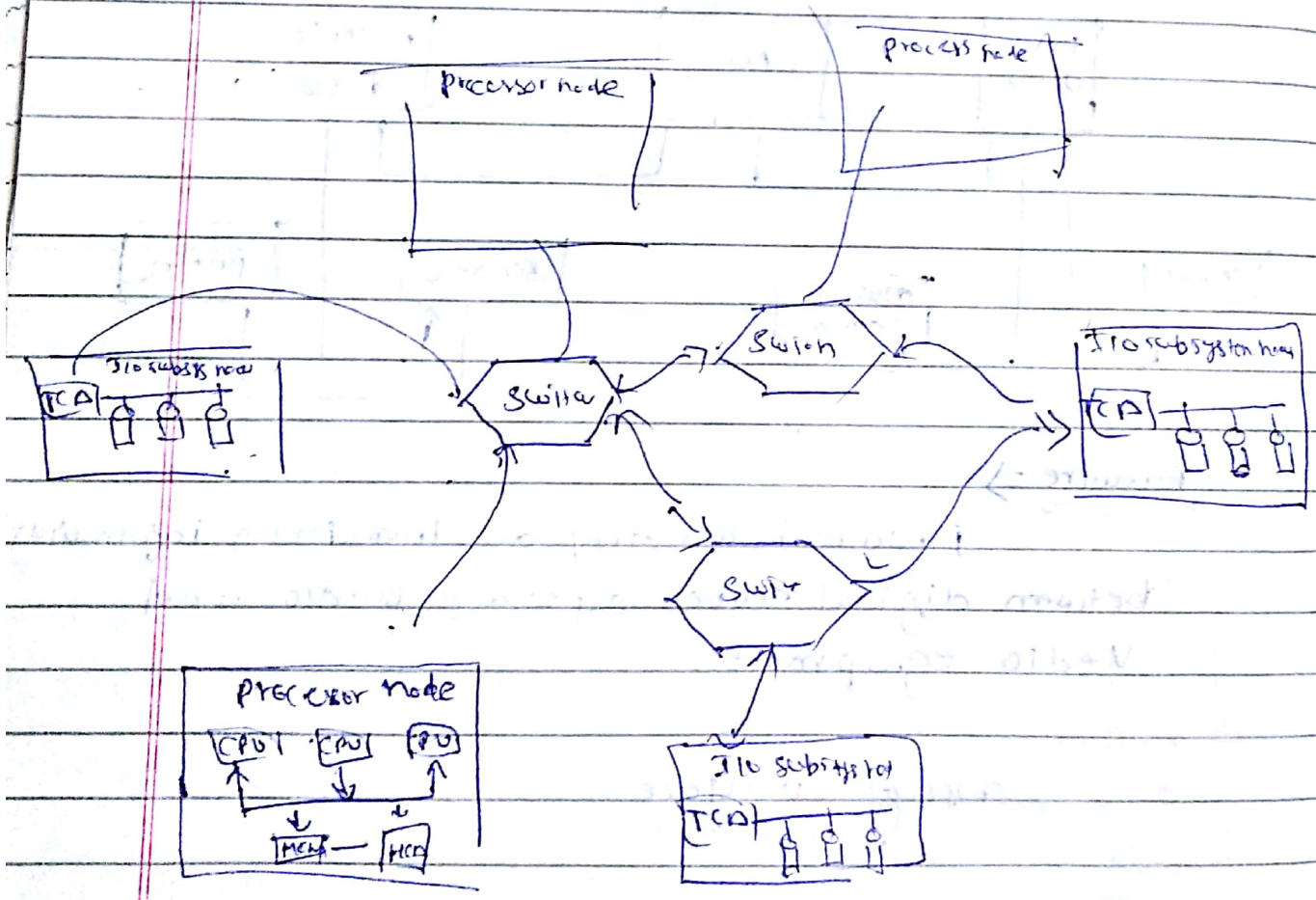
A computer networking communication standard used in high-performance computing.

It is used for data interconnect both among and within computer

⇒ Remote storage, networking and connection between servers

⇒ Scalable data centre

⇒ Independent nodes added as required

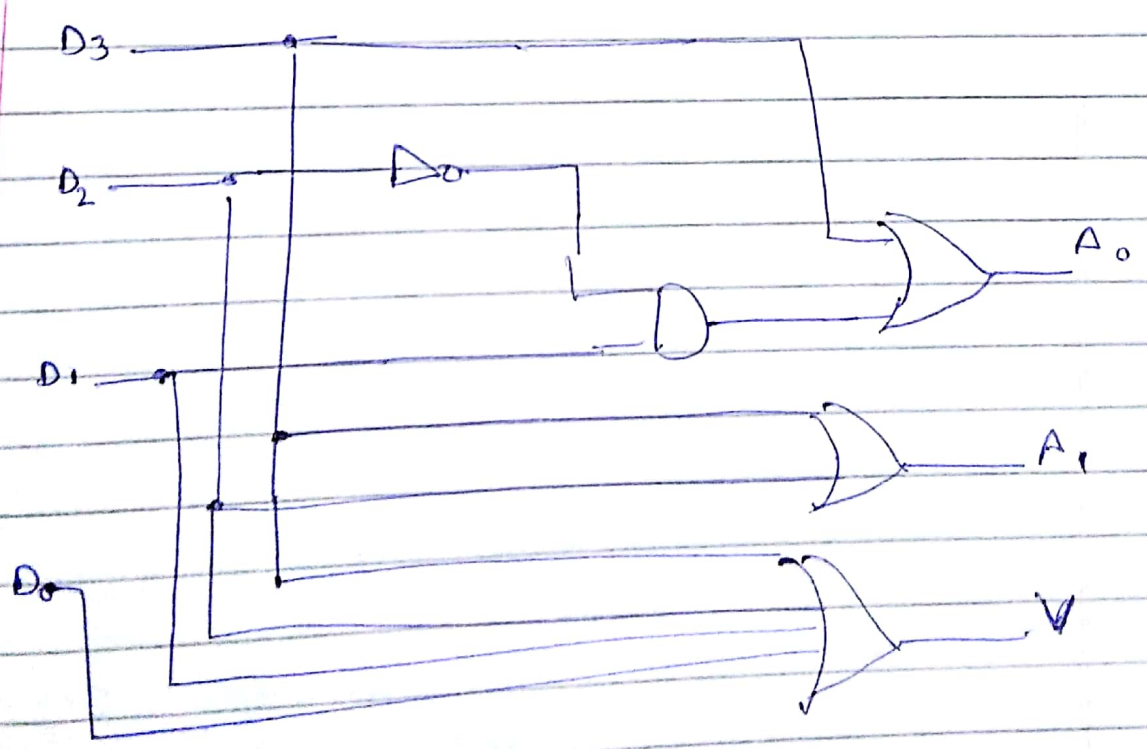


Scanned by CamScanner

4-input priority encoder

Truth table

Input				Output		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1



Scanned by CamScanner