MADE EASY

Computer Science Engineering / IT

Toppers Handwritten Notes

Computer Organization

By-Sagar sir

- Theory
- Explanation
- Derivation
- Example
- Shortcuts
- Previous Years Question With Solution

Computer Organization

10 marks

syllabus:

Module 1: computer architecture

Module 2: computer organization.

Ref Books: 1. computer architecture & organigization.
– Morris Mano. (Hardware design)

2. computer orgn.
– william stallings.

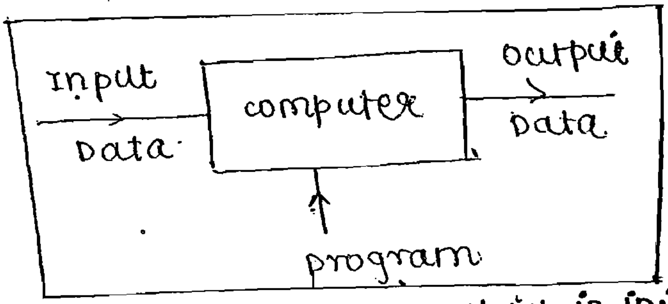
Facuty: Pingili sagar.

email: sagar262003@yahoo.co.in.

keywords:

computer:

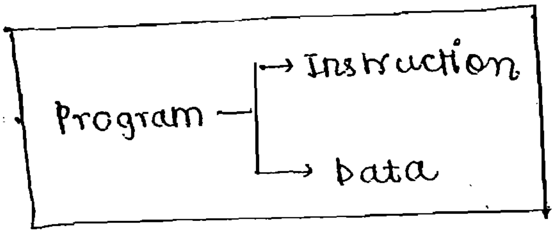
computer is a computational Machine used to process the data. under the control of a application program. Therefore computer system functionality is program execution.



(program which is initiated by user)

program:

Program is a sequence of instructions along with the data.



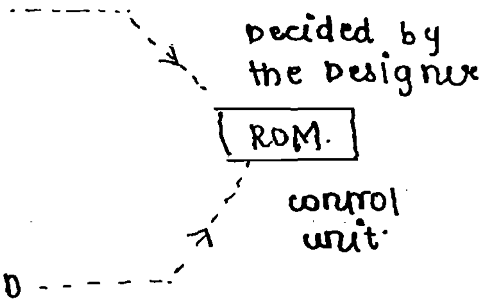
Instruction:

Instruction is a Binary code which is designed inside the processor to confirm some task.

Binary code – Bind with – operation

Eg: If CPU - 'x' supports 8 different operation then opcode = $\log_2 8$ = 3bit.

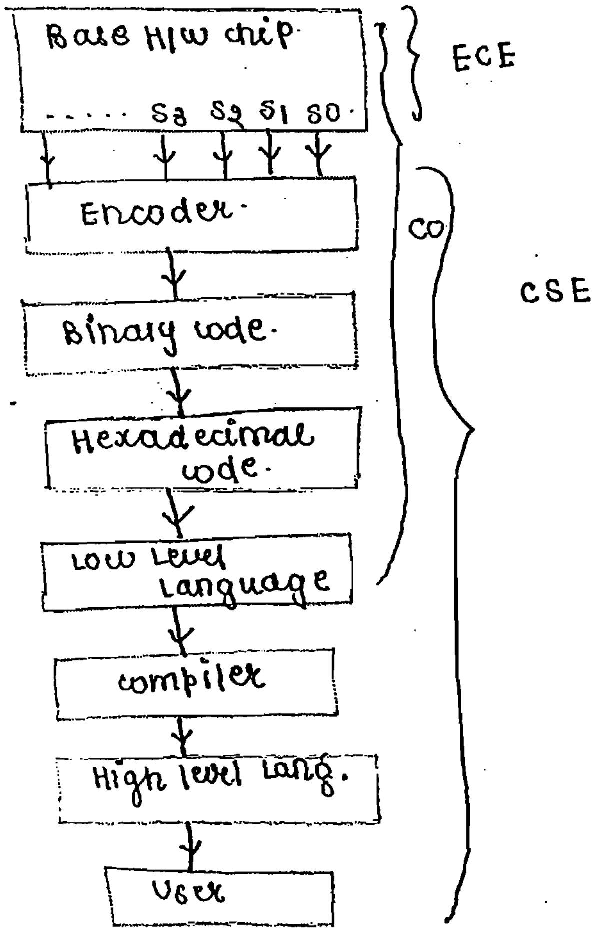
Binary (opcode) code	operation.
000	+
001	−
010	*
⋮	⋮
111	AND



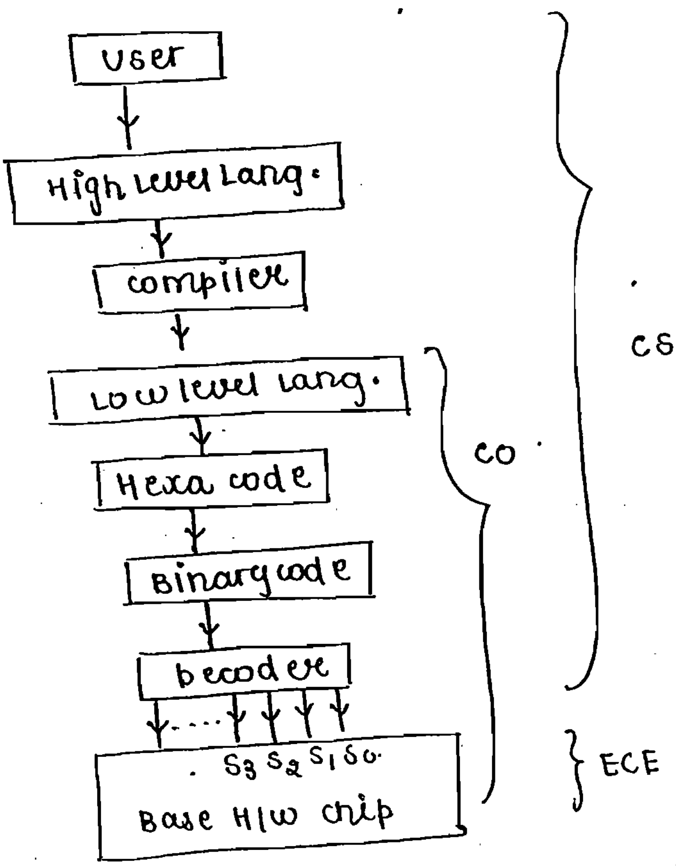
Encoding process : n signals given How many bits required to process signals $\log_2 n$.

Decoding process : n bits are given, How many operation can be performed by computer : '2^n' operation.

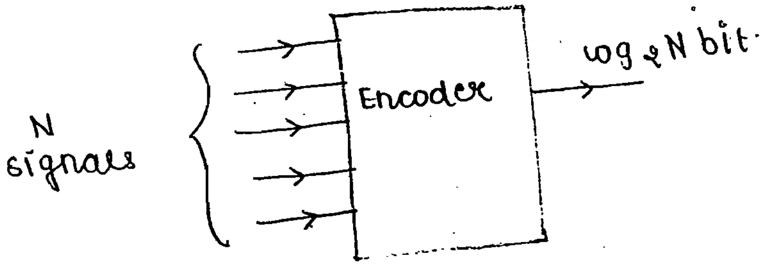
Designer view :



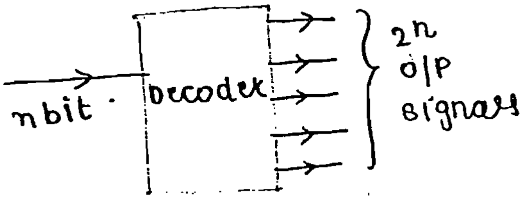
User view :



Encoding: In this process 'N' signals are represented using $\log_2 N$ bit format.



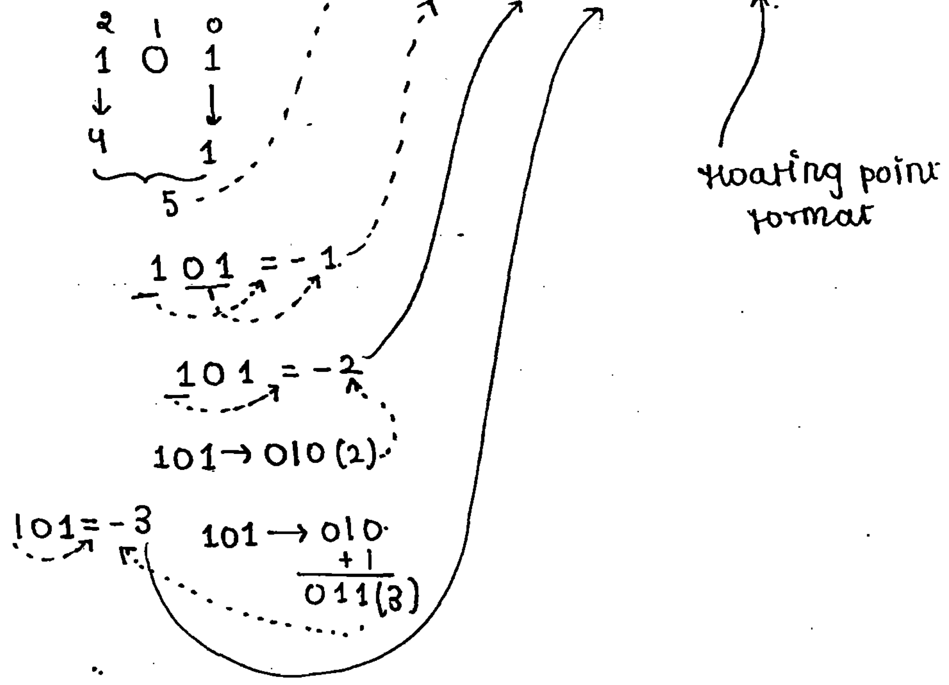
Decoding: In this process, n bit Decoder produces 2^n output signals.



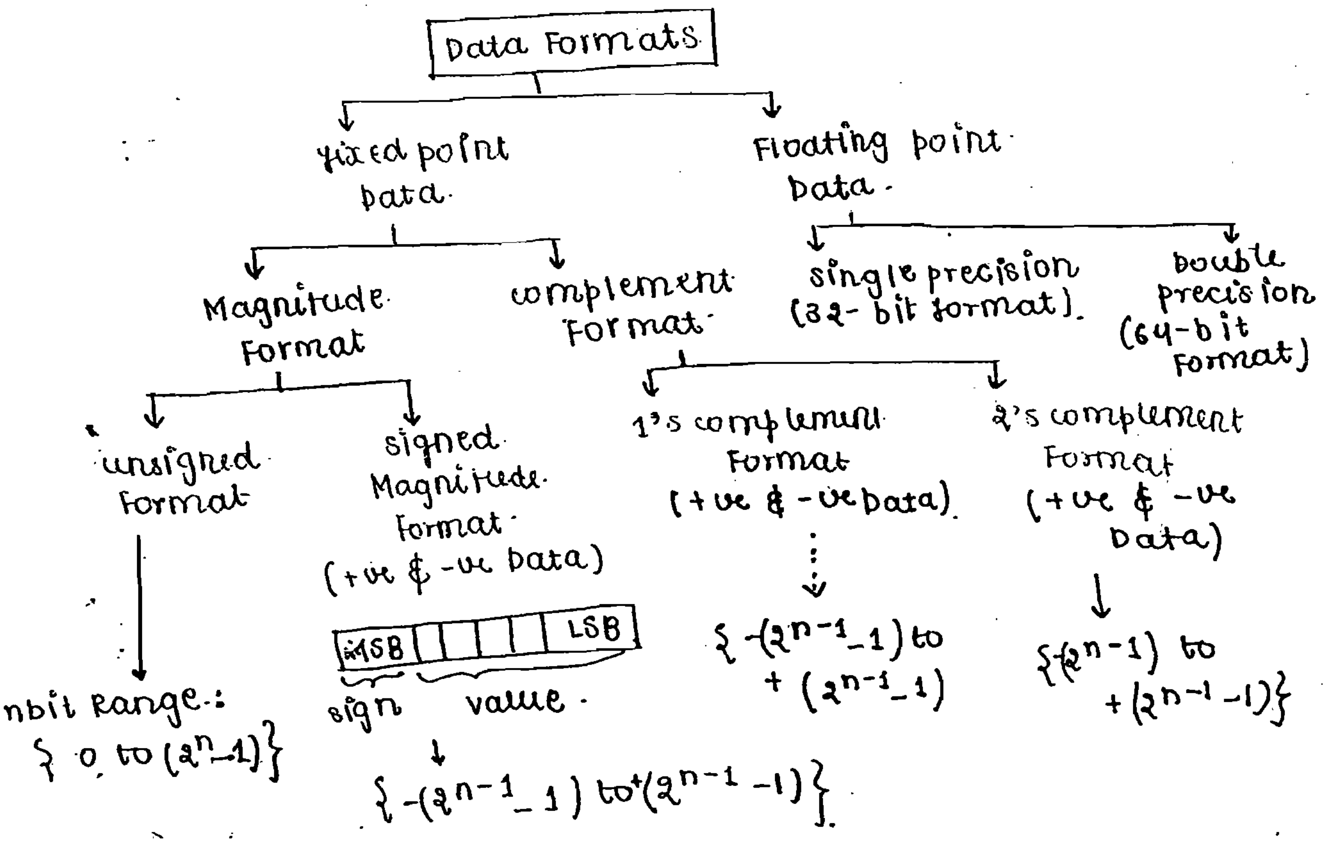
④. <u>Data</u>: It is a Binary code which is associated with a value based on the Data format.

Binary code -- Bind with - value

<u>eg</u>: $(101)_2$ = 5 -1 -2 -3 , fraction



<u>Data Representation</u>:



Data Formats
fixed point Data
Floating point Data
Magnitude Format
complement Format
single precision (32- bit format)
Double precision (64-bit Format)
unsigned Format
signed Magnitude format (+ve & -ve Data)
1's complement Format (+ve & -ve Data)
2's complement Format (+ve & -ve Data)
MSB
LSB
sign
value
nbit Range:
{ 0 to (2^n-1)}
{-(2^{n-1}-1) to +(2^{n-1}-1)}
{-(2^{n-1}-1) to +(2^{n-1}-1)}
{(2^{n-1}) to +(2^{n-1}-1)}

Fixed point Data

4 Bit Binary.	Unsigned Data.	Sign Magnitude.	1's complement	2's complement.
0000	0	$\boxed{+0}$	$\boxed{+0}$	+0
0001	1	+1	+1	+1
0010	2	+2	+2	+2
0011	3	+3	+3	+3
0100	4	+4	+4	+4
0101	5	+5	+5	+5
0110	6	+6	+6	+6
0111	7	+7	+7	+7.
1000	8	$\boxed{-0}$	-7	-8
1001	9	-1	-6	-7
1010	10	-2	-5	-6
1011	11	-3	-4	-5
1100	12	-4	-3	-4
1101	13	-5	-2	-3
1110	14	-6	-1	-2
1111	15	-7	$\boxed{-0}$	-1

Data Redundancy Problem.

"NOT in" 'USE

"NOT in USE"

1's complement

1000 = -7

111(7)

1001 = -6

110(6)

2's complement

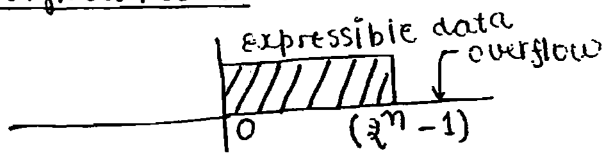
1000 = -8

000
111
+ 1
1000(8)

1001 = -7

110
+1
111(7)

unsigned Data



eg: 4 bit Data {0 to 15}

$$\begin{array}{r r}
 & \boxed{1}\ 1\ 1\ 1 \\
15 & :\ 1\ 1\ 1\ 1 \\
+\,15 & :\ 1\ 1\ 1\ 1 \\
\hline
30 & 1\ 1\ 1\ 0
\end{array}$$

↓ overflow

↓

Test with 5 bit Data : {0 to 31}

NOTE:

(n-bit) + (n bit) = (n+1) bit

↓

1 bit storage space required

↓

1 Flip flop

↓

Flag

↓

carry flag

↓

condition : "Is there an extra bit out of MSB" (or) "Is Borrow required into the MSB"

→ T = set = 1 = C

→ F = reset = 0 = NC

eg:

$$\begin{array}{r r}
 & \times\,1\,1\ \ \\
6 : & 0\,1\,1\,0 \\
\oplus\ 7 & 0\,1\,1\,1 \\
\hline
13. & 1\,1\,0\,1 \\
\hline
CY : 0 & CY : 0
\end{array}$$

Justify PSW (program status word)

↓ Acc → 1101

↓ flag Register → CY = 0

Ans: 1101 (13)

Eg:

```
  8      1000
  9      1001
 ----    ----
 17      0001
 ----    ----
cy=1.    cy=1
```

Justify:

```
              PSW
        ┌──────┴──────┐
        ↓             ↓
       Acc         Flag Reg
        ↓             ↓
      0001          cy=1
Ans:         10001 (17)
```

Multiplication.

- Multiplicatn process is controlled by a Multiplier.
- Two actions are present in the Multiplication.
  (1) generation of partial product
  (2) summation of partial product.
- partial product is generated based on the Multiplier bits. i.e when the multiplier bit is '1' partial product is Multiplicant otherwise partial product is zero. (0).
- After the generation of a partial product, provide the solution to produce the final product.

```
   Multiplicant      Multiplier.
     1 1 1 1     *    1 1 1 1
                     ←── LSB
   ------------------------------
         2 1
       3 1 1 1 1              }
     3 1 1 1 1 x              }  partial products.
   2 1 1 1 1 x x              }
   1 1 1 1 x x x
  -------------------------------
  1 1 1 0 0 0 0 1               } final product
  -------------------------------
```

2 → 10 carry(1).

4 → 100 carry(2).

6 = 110 carry(3)

5 = 101 carry(2).

NOTE:

(nbit) * (nbit) = 2n bit.

↓

Register pair is used to report the result

Ques: consider the following Multiplication.

$(10w1Z)_2 * (15)_{10} = (Y01011001)_2$.

what are the value of w, Y, & z variables?

$(15)_{10} = (1111)_2$.

$$\begin{array}{r}
10w1Z \quad * \quad 1111. \\ \hline
{}^{2\;2\;1} \qquad \\
{}^{2}\,10w\,1Z. \\
{}^{2}\,10\cancel{w}\,1\,Zx \\
1\;10\cancel{w}1\cancel{Z}x\,x \\
10\cancel{w}_1 1 Z_1 x\,x\,x \\ \hline
\cancel{1}01011001 \Rightarrow (Y01011001)_2.
\end{array}$$

y=1

$\boxed{z=1}$

$1+Z \Rightarrow 1+1=\underline{\underline{0}}$ 1 carry.

Now replace z with 1.

Now; if (w=0). for $\boxed{1+w+1+1=0}$

1+0+1+1 = 3(11) Not matching

if (w=1)

1+1+1+1 = 4 (100) matching.

↳ carry(2)

$\boxed{w=1}$

NOTE: In a Mannual Multiplication process, 2 limitations present

(1) Requires More Registers to Hold the partial product

(2) summation process become complex in the H/w Therefore optimization Required that is Accumulated addition. Described in Flow chart