DE:

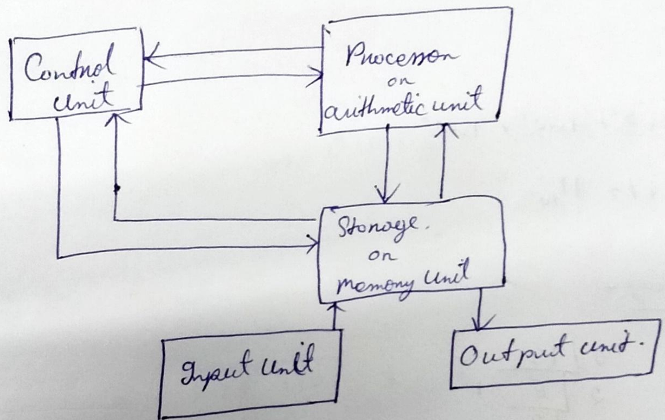
Unit-1: Ch-1:

Digital Computer: A digital computer is an electronic device that process information using binary numbers (0s and 1s)

Components:

① Input Unit ② Processing Unit ③ Memory unit ④ Output unit

⑤ Storage Unit.



Types:

(i) Supercomputers (ii) Mainframes (iii) Minicomputer (iv) Microcomputers

(v) Embedded Computers.

Digital system: It is a system that processes digital signals. It consists of Logic circuits, memory and processing units.

Binary number: It is a number system that uses only two digits 0 and 1.

- Digital computers and electronic circuits work with two voltage levels (ON = 1, OFF = 0);
- Easier to process and store compared to other number system.

Represented by - $\boxed{\text{bit}}$

Conversion -

Binary to decimal:

$1011_2 \rightarrow$

$1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= 8+0+2+1 = 11_{10}$

Decimal to binary -

$13_{10} \rightarrow$

2	13	
2	6	1
2	3	0
	1	1

$= 1101_2$

Binary arithmetic:

① Addition:

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 1 = 10 (carry 1)
- 1+1+1 = 11 (carry 1)

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

② Subtraction -

- 0 - 0 = 0
- 0 - 1 = 1 with 1 borrow
- 1 - 0 = 1
- 1 - 1 = 0

$$\begin{array}{r} 1101 \\ -\ 1011 \\ \hline 0010 \end{array}$$

Number base Conversion:

① Binary to decimal -

$1011_2 \rightarrow 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 = 8+0+2+1 = 11_{10}$

② Decimal to Binary -

$13_{10} \rightarrow$

$$\begin{array}{r|ll} 2 & 13 & \\ 2 & 6 & 1 \\ 2 & 3 & 0 \\ & 1 & 1 \end{array} = 1101_2$$

③ Binary to Octal -

3 bit sets

$101110_2 \rightarrow \overline{101}\ \overline{110} \Rightarrow 101 = 5 \quad 110 = 6 \quad = 56_8$

④ Octal to Binary -

$27_8 \Rightarrow 2 = 010 \Rightarrow 010111_2$

$7 = 111$

⑤ Binary to Hexadecimal -

4bit

$10101101_2 = \overline{1010}\ \overline{1101} = A\,D_{16}$

⑥ Hexa to Binary -

$B3_{16} = B = 1011_2 = 10110011_2$

$3 = 0011_2$

⑦ Decimal to Octal:

$156_{10} =$

$$\begin{array}{r|ll} 8 & 156 & \\ 8 & 19 & 4 \\ & 2 & 3 \end{array} = 234_8$$

⑨ Decimal to Hexa:

$255_{10} \rightarrow$

$$\begin{array}{r|ll} 16 & 255 & \\ & 15 & 15 \end{array} = FF_{16}$$

6

Complements in Digital System :

⇒ 1's & 2s Complement

$(110101)_2$

⇒ Replace 1 with 0 & 0 with 1.

001010 ← 1's complement

+ 1 add +1 to get 2's complement

$(001011)_2$ ← 2's complement.

1011010

⇒ 0100101

+ 1

0100110

⇒ 7's & 8's Complement in Octal Number System

$(67123)_8$

77777

− 67123

10654

⇒ 10654 → 7's Complement

to get 7's compliment of given Octal NS we need to subtract each number from 7.

10654 add 1 to get 8's compliment.

+ 1

10655 → 8's

⇒

9's & 10's Compliment in Decimal NS.

$(169700)_{10}$ to get 9's compliment, we need to subtract from 9

999999

− 169700

830299 → 9's compliment.

+ 1

830300 → 10's compliment.

(543)$_{10}$

$$\begin{array}{r} 999 \\ -\,543 \\ \hline 456 \\ +\,1 \\ \hline 457 \end{array}$$

15's and 16's Compliment

$(2BCDE)_{16}$

$$\begin{array}{rrrrr} 15 & 15 & 15 & 15 & 15 \\ -\,2 & B & C & D & E \\ \hline D & 4 & 3 & 2 & 1 \end{array}$$ — 15's compliment

$$\begin{array}{rrrrr} & & & + & 1 \\ \hline D & 4 & 3 & 2 & 2 \end{array}$$ → 16's compliment

BCD		Excess-3 Code
0 - 0000	+0011 →	0011
1 - 0001	→	0100
2 - 0010	→	0101
3 - 0011	→	0110
4 - 0100	→	0111
5 - 0101	→	1000
6 - 0110	→	1001
7 - 0111	→	1010
8 - 1000	→	1011
9 - 1001	→	1100

Binary storage and Registers:

① Binary storage: Refers to device or circuits that store binary data (0s and 1s)

ⓐ Volatile Storage
⇒ Loses data when Power is OFF
eg- RAM, CPU registers

ⓑ Non Volatile Storage.
⇒ data remains when power is OFF.
- ROM, SSD.

Basic storage unit - Flip Flops.
↳ Basic building blocks of storage.
↳ Single flip flop store one bit at a time (0 or 1)
↳ Multiple flip-flops forms register and memory units.

② **Registers:** Is a group of flip-flops used to store multiple bit. Registers are used in the CPU for temp. data storage and fast processing.

Types.

① Data Register (D R) → Stores data processing.

② Accumulator → Stores intermediate arithmetic results.

③ Instruction Register → Holds the current instruction.

④ Program Counter → Store the address of next instruction.

⑤ Stack Pointer → Points to the top of the stack in memory.

③ **Shift Registers:** Special type of register that shifts data left or right on each clock pulse.

- Used in serial data transfer, encryption and digital circuits.

Binary Logic:

- 0 (Low, False, OF)
- 1 (High, True, ON)

AND (.)

A	B	A · B
0	0	0
0	1	0
1	0	0
1	1	1

OR (+)

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

NOT (Inversion)

A	A'
0	1
1	0

② Universal Gates.

ⓐ NAND Gate.

A	B	A NAND B($\overline{A}\cdot B$)
0	0	1
0	1	1
1	0	1
1	1	0

NOR

A	B	$\overline{A+B}$
0	0	1
1	0	0
0	1	0
1	1	0

XOR

A	B	$A\oplus B$
0	0	0
0	1	1
0	1	1
1	1	0

XNOR

A	B	$\overline{A\oplus B}$
0	0	1
1	0	0
0	1	0
1	1	1

Boolean algebra and Logic simplification

Boolean Th^m

- Idempotent Law - $A+A = A$, $A\cdot A = A$

  Complement Law - $A+\bar{A} = 1$, $A\cdot\bar{A} = 0$

  Absorption Law - $A+(A\cdot B) = A$

- DeMorgan's Th^m -
  - $\overline{A+B} = \bar{A}\cdot\bar{B}$
  - $\overline{A\cdot B} = \bar{A}+\bar{B}$

Laws of BA

- Commulative law – Interchanging order of ~~operators~~ operands in a boolean equation doesnot change its result.

  - OR → $A + B = B + A$
  - AND → $A * B = B * A$.

- Associative law – $(A * B) * C = A * (B * C)$

- Distributive law – Multiplication of two variables and adding the result with a variable will result in the same value as multiplication of addition of ~~variation~~ variable with individual variable.

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

- Annulment – $A \cdot 0 = 0 \Rightarrow A + 1 = 1$

- Identity law – $A \cdot 1 = A \Rightarrow A + 0 = A$

- Idempotent law – $A + A = A \Rightarrow A \cdot A = A$

- Complement law $A + A' = 1$ & $A \cdot A' = 0$

- Double negation law – $((A)')' = A$

- Absorption law – $A \cdot (A + B) = A$

  $A + AB = A$

- De Morgans – $(A \cdot B)' = A' + B'$

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