UNIT-IV

Digital Electronics fondamentals

Advantage of digital System over analog →

1) Digital Circuit is easy to design or implement
2) It gives high accurate value.
3) Processing speed & transmission rate of the information is faster.
4) The information related to digital ckt is less affected by external disturbance or noise.
5) Power requirement is less.
6) Information storage is easy.

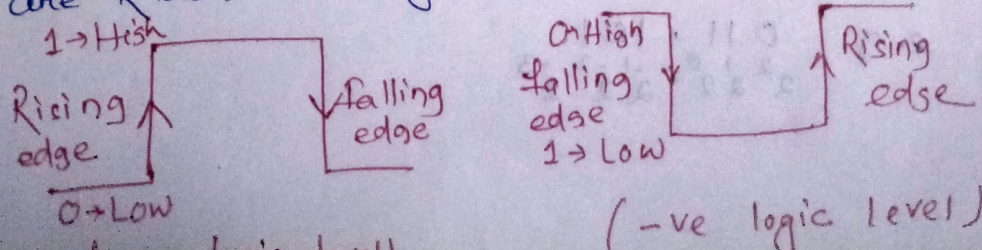
Binary digit → In a digital system, two digits are used to transmit the data which is known as binary digit. It's either 0 or 1.

↳ Zero represents low value &
One " high.

↳ Binary digits are known as bits and group of bits known as codes.

Logic Levels → Two logic levels are used:
a) Positive b) Negative
a) In +ve level, 0 represents low value & 1 represents high.
b) In -ve level, 0 represents high & 1 represents low value.

* The voltages used to represent 0 & 1 are known as logic levels.



UNIT-IV

Digital Electronics fondamentals

Advantage of digital System over analog →

1) Digital circuit is easy to design or implemen
2) It gives high accurate value.
3) Processing speed & transmission rate of the information is faster.
4) The information related to digital ckt is less affected by external disturbance or noise.
5) Power requirement is less.
6) Information storage is easy.

Binary digit → In a digital system, two digits are used to transmit the data which is known as binary digit. It's either 0 or 1.

↳ Zero represents low value &
One " high.

↳ Binary digits are known as **bits** and group of bits known as **codes**.

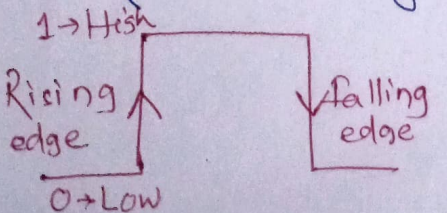
Logic Levels → Two logic levels are used:

a) Positive b) Negative

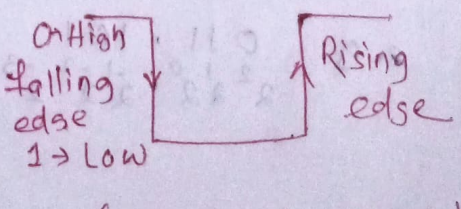
a) In +ve level, 0 represents low value & 1 represents high.

b) In -ve level, 0 represents high & 1 repres
low value.

* The voltages used to represent 0 & 1 are known as **logic levels**.



(+ve logic level)



(-ve logic level)

Number System → It is the language of representation of data consisting of a ordered set of symbols known as digits.

* It's of 4 types :-
  i) Decimal
  ii) Binary
  iii) Octal
  iv) Hexadecimal

i) Decimal number system →

a) Maximum limit is 10, which is known as base or radix. It specifies the actual no. of digits included in number system.

b) 0 to 9.

c) It's a positional weighted system, it means each digit has some value depends on the location w.r.t decimal point.

for eg, 1 2 3 · 5 7
d^2 d^1 d^0 d^{-1} d^{-2}
10^2 10^1 10^0 · 10^{-1} 10^{-2} → Decimal

ii) Binary Number System →

a) Base is 2 = 0, 1

b) It's a positioned weighted system in which the weight of each successive higher position to the left is an increasing power of 2.

c) A binary digit is known as bit.

e.g. 0 1 1 · 1 0 1
2^2 2^1 2^0 · 2^{-1} 2^{-2} 2^{-3}

2

iii) Octal number System →

↳ Base is 8 - 0 to 7.

↳ It has 8 independent symbols.

↳ Since its base $8 = 2^3$, every 3 bit group of binary can be represented by an octal digit.

↳ e.g $(2\ 3\ 5)_8$
$8^2\ 8^1\ 8^0$

octal	Binary ($2^2\ 2^1\ 2^0$)
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

iv) Hexadecimal →

↳ Base is 16 Hexa (16)

↳ It has 16 independent symbols i.e 0 to 15

0 to 9, A B C D E F
10 11 12 13 14 15

↳ Binary $16 = 2^4$.

To convert the hexadecimal number into binary make a group of 4 digits because $16 = 2^4$.

Hexa(16)

	Binary 2^3 2^2 2^1 2^0	
0 →	0 0 0 0	0
1 →	0 0 0 1	1
2 →	0 0 1 0	2
3 →	0 0 1 1	3
4 →	0 1 0 0	4
5 →	0 1 0 1	5
6 →	0 1 1 0	6
7 →	0 1 1 1	7
8 →	1 0 0 0	8
9 →	1 0 0 1	9
10 →	1 0 1 0	A
11 →	1 0 1 1	B
12 →	1 1 0 0	C
13 →	1 1 0 1	D
14 →	1 1 1 0	E
15 →	1 1 1 1	F

Converesions →

① Decimal to binary →

Divide the number by 2 & write the reemainder and consider the answer from bottom to top.

But if it's fraactional value, multiply the number by 2 & consider the remainder from top to bottom.

e.g $(48)_{10}$

2 | 48 — 0
2 | 24 — 0
2 | 12 — 0
2 | 6 — 0
2 | 3 — 1
1

$(48)_{10} = (110000)_2$

(3)

e.g ≡ ~~$(163)_{10}$~~ Convert $(163.875)_{10}$ to binary.

		remainder
2 | 163		
2 | 81		1
2 | 40		1
2 | 20		0
2 | 10		0
2 | 5		0
2 | 2		1
2 | 1		0
0		1

$(163)_{10} = (10100011)_2$

$(0.875)_{10}$ =

$0.875 \times 2 = 1.75$ 1

$0.75 \times 2 = 1.5$ 1

$0.5 \times 2 = 1.0$ 1

$(163.875)_{10} = (10100011.111)_2$

(2) ~~Binary to~~ Decimal to octal →

e.g ① Convert $(378.93)_{10}$ to octal.

8 | 378		
8 | 47	2	
5	7	

$0.93 \times 8 = 7.44$ 7

$0.44 \times 8 = 3.52$ 3

$0.52 \times 8 = 4.16$ 4

$0.16 \times 8 = 1.28$ 1

$(378.93)_{10} = (572.7341)_8$

② $(128.625)_{10}$

$0.625 \times 8 = 0.5$ 0

8 | 128	
8 | 16	0
2	0

$(128.625)_{10} = (200.5)_8$

③ Decimal to hexadecimal :-

1) Convert $(52.675)_{10}$ to its hexadecimal form.

16 | 52
3 4

0.675 × 16 = 10.8 → 10 A
0.8 × 16 = 12.8 → 12 C
0.8 × 16 = 12.8 → 12 C

$(52.675)_{10} = (34.ACC\ldots)_{16}$

$= (34.A\bar{C})_{16}$

2) Convert $(2598.675)_{10}$ to hexadecimal.

16 | 2598
16 | 162 → 6
16 | 10 → 2
0 → 10 (A)

0.675 × 16 = 10.8 → 10 (A)
0.8 × 16 = 12.8 → 12 (C)
0.8 × 16 = 12.8 → 12 (C)
0.8 × 16 = 12.8 → 12 (C)

$(2598.675)_{10} = (A26.ACCC\ldots)_{16}$

4) Binary to decimal :-

* Multiply each bit with the corresponding power of 2 & then add all the digit.

e.g) (1) Convert $(10101)_2$ to decimal.

$10101 = (1\times2^4)+(0\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$
$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$= 16+4+1 = (21)_{10}$

(2) $(110\ 11.\ 101)$ to decimal.
$2^4\ 2^3\ 2^2\ 2^1\ 2^0\ .\ 2^{-1}\ 2^{-2}\ 2^{-3}$

$(1\times2^4)+(1\times2^3)+(0\times2^2)+(1\times2^1)+(1\times2^0)+(1\times2^{-1})+(0\times2^{-2})+(1\times2^{-3})$

$= 16+8+0+2+1+\frac{1}{2}+\frac{1}{8}$

$= (27.625)_{10}$

4

10) Octal to hexadecimal ⇒

(8) Octal to decimal ⇒

1) Convert $(4057.06)_8$ to decimal.

4057.06 (positions $8^3\ 8^2\ 8^1\ 8^0 . 8^{-1}\ 8^{-2}$)

$= (4\times8^3)+(5\times8^1)+(7\times8^0)+(6\times8^{-2})$

$= 2048 + 40 + 7 + 0.0937$

$= (2095.0937)_{10}$

2) Convert octal number 726 into its decimal equivalent.

$(726)_8 = (7\times8^2)+(2\times8^1)+(6\times8^0)$

$= 448 + 16 + 6$

$= (470)_{10}$

3) $(12.2)_8 = (?)_{10}$

12.2 ($8^1\ 8^0 . 8^{-1}$)

$= (1\times8^1)+(2\times8^0)+(2\times8^{-1})$

$= 8 + 2 + 0.25$

$= (10.25)_{10}$

(8) Hexadecimal to decimal ⇒

1) Convert $(5C7)_{16}$ to decimal.

$5C7$ ($16^2\ 16^1\ 16^0$)

$= (5\times16^2)+(C\times16^1)+(7\times16^0)$

$= 1280 + 192 + 7$

$= (1479)_{10}$

2) Convert $(A0F9.0EB)_{16}$ to decimal

$A0F9.0EB = 10\ 0\ 15\ 9 . 0\ 14\ 11$ ($16^3\ 16^2\ 16^1\ 16^0 . 16^{-1}\ 16^{-2}\ 16^{-3}$)

$= (10\times16^3)+(0\times16^2)+(15\times16^1)+(9\times16^0)+(0\times16^{-1})+(14\times16^{-2})+(11\times16^{-3})$

$= 40960 + 0 + 240 + 9 + 0.0546 + 0.0026$

$= (41209.0572)_{10}$

7) Binary to octal ⇒

↳ Binary numbers can be converted into equivalent octal no. by making groups of 3 bits starting from LSB & moving towards MSB.

Convert $(110101.101010)_2$ to octal.

110 101 . 101 010

6 5 5 2

$(65.52)_8$

Convert $(10101111001.0111)_2$ to octal.

010 101 111 001 . 011 100 = $(2571.34)_8$

2 5 7 1 3 4

8) Binary to hexadecimal ⇒

↳ Make a group of 4 bit & add no. of zeros if required.

Convert $(1011011011)_2$ to hexadecimal.

0010 1101 1011 = $(2DB)_{16}$

2 D B

Convert $(0101111111011.011111)_2$ to hexa.

0010 1111 1011 . 0111 1100 = $(2FB.7C)_{16}$

9) Octal to binary ⇒ $(\ )_8 = (\ )_2$

Convert $(367.52)_8$ to binary.

367.52 = 011 110 111 . 101 010

$(011110111.101010)_2$

↳ Replace each octal digit by its 8-bit binary equivalent.

4. 10) Octal to hexadecimal →

To convert ~~hexadecimal no~~ octal no. to hexadecimal no, represent each bit to its equivalent 4 digit binary no.

Convert $(756.603)_8$ to hexadecimal.

7	5	6	.	6	0	3
0111	0101	0110	.	0110	0000	0011

11) Hexadecimal to octal →

First convert the given hexa. number to binary & then the binary to octal.

Convert $(B9F.AE)_{16}$ to octal.

B	9	F	.	A	E
1011	1001	1111		1010	1110

101110011111 . 10101110

$(5637.534)_8$

Binary Arithmetic →

1) Binary addition
2) Binary Substraction
3) Binary Multiplication
4) Binary Division

1) Binary addition →

Rules :-

	Sum	Carry
0+0 =	0	0
0+1 =	1	0
1+0 =	1	0
1+1 =	0	1
1+1+1 =	1	1

... rules are

Sum → Result of an addition operation
Carry → A bits that holds the value of 0 & 1.

Add, binary numbers 1010 & 111.

```
  1010
   111
 10001
```

Add, binary numbers 1101.101 & 111.011.

```
  1101.101
   111.011
 10101.000
```

Binary Subtraction →

Rules :-

0 - 0 = 0

1 - 1 = 0

1 - 0 = 1

0 - 1 = 1, with a borrow of 1.

Borrow :- A digit brought back from a more significant position when the subtracted digit.

Subtract 10_2 from 1000_2.

```
  1000
    10
  0110
```

Subtract $(111.111)_2$ from $(1010.01)_2$

```
  1010.01
- 111.111
  0011.
```

Complement :-

To Convert a number into its 1's complement form, only change all the Zero into 1 & 1 into 0.

* If the number is +ve, the magnitude is represented in its true binary form and a sign bit 0 is placed in front of MSB i.e left most bit.

* If the number is -ve, the magnitude is represented in its 2's (or 1's) complement form & a sign bit 1 is placed in front of MSB.

e.g find 1's complement & 2's complement of 7.

Binary represent^n of 7 = 1 1 1

1's complement = 0 0 0

2's complement = 0 0 0
+ 1
---
0 0 1

Sign magnitude number ⇒

* If a number is represented by a sign magnitude number, then the left most bit represents the sign bit which specifies whether the no. is +ve or -ve.

* e.g $+12 = (1100)_2$

8-bit representation →

$+12 = (00001100)_2$

Sign (+) | magnitude

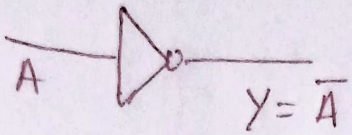
* If the sign bit is 0, then the no. is +ve else -ve no.

Logic gates :- Logic gates are used to carry out logical operations on single or multiple binary inputs and give one binary output.

There are several basic logic gates used in performing operations in digital systems.

Examples :-

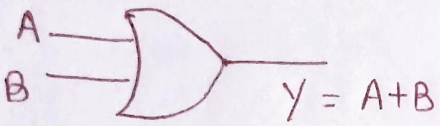
① NOT gate :-



↳ Truth table

A	Y = Ā
0	1
1	0

② OR

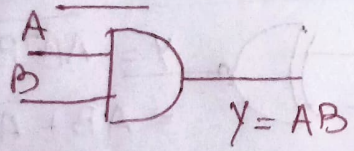


↳

i/p A	i/p B	output Y = A+B
0	0	0
0	1	1
1	0	1
1	1	1

The o/p attains 1 if one or more inputs attain 1.

③ AND



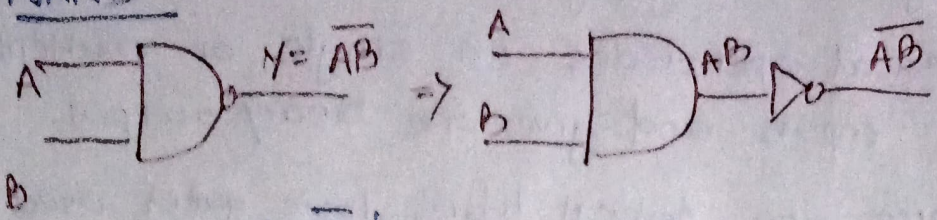
A	B	Y = AB
0	0	0
0	1	0
1	0	0
1	1	1

→ The o/p is 1 if and only if all the i/ps are 1.

Universal gates :-

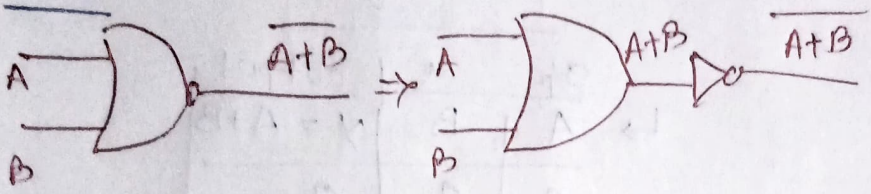
There are 2 types :- a) NAND b) NOR

NAND



A	B	$Y = \overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

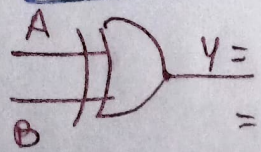
NOR



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

Some basic gates cont —

Exclusive-OR (XOR)

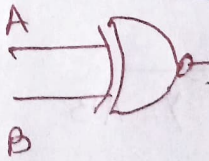


$Y = A\overline{B} + \overline{A}B$

$= A \oplus B$

A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive NOR (XNOR)



$Y = \overline{A \oplus B}$

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

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