S.No.	Date	Title	Page No.	Teacher's Sign/ Remarks
1.		Computer Fundamental * Number system * Logic gate * Codes * Overview of computer organization and architecture		

[Module NO : 01]

[Number system]

Number system :-

* A Number system is method to represent the number *

(OR)

* The number system is used for representing a quantity or information in digital electronics

Digit :- In any number system, every symbol in the number is called a digit.

MSD [Most significant Digit] :-

The Leftmost digit of the number has the gratest positional weight among the remaining digit in number. It is called as Most significant digit in number. It is called as Most Significant Digit (MSD)

LSD [Least significant Digit] :-

The rightmost digit of the number has the least positional weight among the digit in number. It is called as Least significance Digit (LSD)

MSB LSB

$$7 \quad 2 \quad 8 \quad . \quad 1 \quad 4$$

$$7 \times 10^2 \quad 2 \times 10^1 \quad 8 \times 10^0 \quad 1 \times 10^{-1} \quad 4 \times 10^{-2}$$

Base / Radix :-

The base or radix (b) of the number system is the total number of digit used in that number system

example:- Binary system → (0,1)

Base / Radix = 2

Types of number system:-

i) Binary number system

The number system with Base/ Radix is 2, then it is called as binary number system.

OR

A number system where a number system

Types of Numbersystem

By Rutik Bod

→ Binary :- (0,3) $\boxed{\text{Base} = 2}$

→ octal (0, 1, 2, 3, 4, 5, 6, 7) $\boxed{\text{Base} = 8}$

→ Decimal (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) $\boxed{\text{Base} = 10}$

→ Hexa Decimal $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \overset{A}{\cancel{10}}, \overset{B}{\cancel{11}}, \overset{C}{\cancel{12}}, \overset{D}{\cancel{13}}, \overset{E}{\cancel{14}}, \overset{F}{\cancel{15}})$

$\boxed{\text{Base} = 16}$

$$\text{Base} \rightarrow 2 \overline{)\,4\,}\ \ \begin{array}{l} 2 \leftarrow \text{Quotient} \\ -4 \\ \hline 0 \leftarrow \text{remainder} \end{array}$$

Noted that

Noted that

Base	Quotient	
2	199	
2	99	→ 1
2	49	→ 1
2	24	→ 1
2	12	→ 0
2	6	→ 0
2	3	→ 0
2	1	→ 1

} → remainders

* Take calculator
* Divided that number $2 \div 13 = 6.5$ → 6
* multiply that number $0.5 \times 2 = 1.0$ → 1

2	13
2	06 → 01

* fractional part मध्ये 4 वेळा multiplication केला जातो.

Universal college of Engine-
Rutik Bodke
div: A

1

Conversion:-

A) Decimal to Binary

① $(8)_{10} \rightarrow (\ )_2$

2	8	
2	4	→ 0 LSB
2	2	→ 0
2	1	→ 0
	0	→ 1 MSB

Ans:- $(1000)_2$

② $(19.32)_{10} \rightarrow (\ )_2$

2	19	LSB
2	9	→ 1
2	4	→ 1
2	2	→ 0
2	1	→ 0
	0	→ 1 MSB

ANS:-
10011. 0101

① $0.35 \times 2 = \boxed{0}.70$ → 0 MSB
② $0.70 \times 2 = \boxed{1}.40$ → 1
③ $0.40 \times 2 = \boxed{0}.80$ → 0
④ $0.80 \times 2 = \boxed{1}.60$ → 1 LSB

Noted that :-
In fractional part, you can calculate only 4 times

UCOE collage
By Rutik Bodke
div: A

B) Decimal to octal conversion:-

Q.1) $(151.33)_{10} \rightarrow (\quad)_8$

Ans:-

8	151
8	18 → 7 LSB
8	2 → 2
	0 → 2 MSB

$0.33 \times 8 = \boxed{2}.64 \rightarrow 2$ MSB

$0.64 \times 8 = \boxed{5}.12 \rightarrow 5$

$0.12 \times 8 = \boxed{0}.96 \rightarrow 0$

$0.96 \times 8 = \boxed{7}.68 \rightarrow 7$ LSB

ANS:- $(227.2507)_8$

UCOE collage
By Rutik Bodke
div: A

C) Decimal to Hexadecimal conversion

$(151.33)_{10} \rightarrow (\quad)_{16}$

16	151
16	9 → 7 LSB
	0 → 9 MSB

$0.33 \times 16 = \boxed{5}.28 \rightarrow 5$ MSB

$0.28 \times 16 = \boxed{4}.48 \rightarrow 4$

$0.48 \times 16 = \boxed{7}.68 \rightarrow 7$

$0.68 \times 16 = \boxed{10}.88 \rightarrow 10$ A LSB

$0.88 \times 16 = \boxed{14}.08 \rightarrow E$

ANS:-

~~$(155.33)_{10} \rightarrow (97.547E)_{16}$~~

$(155.33)_{10} \rightarrow (97.547AE)_{16}$

2

d) Binary to Decimal conversion

n=4
n-1=3

$(1001.0101)_2 \longrightarrow (\quad)_{10}$

$= 1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2} + 0\times 2^{-3} + 1\times 2^{-4}$

$= 8 + 0 + 0 + 1 + 0 + \frac{1}{2^2} + 0 \frac{1}{2^4}$

$= (8.3125)_{10}$

$\boxed{\text{ANS:- } (1001.0101) \longrightarrow (8.3125)_{10}}$

e) Binary to octal conversion :-

Noted that

For 3 bit	For 4 bit
421	8421

① $(010\ 110\ 010\ 101)_2$

Ans:- $(2\ 6\ 2\ 5)_8$

421
101
010
110
010

$\boxed{\text{Ans:- } (010110010101)_2 \longrightarrow (2625)_8}$

f) Binary to Hexadecimal :-

$(1101\ 1011\ 0101)_2 \Rightarrow (DB5)_{16}$

~~13~~ ~~11~~ 5
D B

8421
0101
1011
1101

3

g) Octal to Binary conversion

$(726)_8 \longrightarrow (111010110)_2$

111 010 110

h) Octal to Decimal conversion

n=3
(n-1) = 3-1 = 2

① $(254)_8 \longrightarrow (\quad)_{10}$

$= (2\times 8^2) + (5\times 8^1)$

i) Octal to Hexadecimal Refer from book for more exploration

$(670.17)_8 \longrightarrow (\quad)_{10}$

00 110 111 000 . 001 111 00 ← make 3 group

← make 4 group

1 B 8 . 3 C

Ans:- $(670.17)_8 \longrightarrow (1B8.3C)_{16}$

4

Hexadecimal to Binary

① $(3A9D)_{16} \rightarrow (\quad)_2$

0011 1010 1001 1101

$\boxed{\text{Ans:- } 3A9D \rightarrow 0011\ 1010\ 1001\ 1101}$

Hexadecimal to octal

① $(BC2.4C)_{16} \rightarrow (\quad)_8$

1011 1100 0010 0100 1100 ← vacate in 4 bits

5 7 0 2 . 2 3 0

Ans:- $(BC2.4C)_{16} \rightarrow (5702.230)_8$

③ Hexadecimal to Decimal

$(4C8.2)_{16} \rightarrow (\quad)_{10}$

$= (4\times16^2)+(12\times16^1)+(8\times16^0)+(2\times16^{-1})$

$= 1024+192+8+0.125$

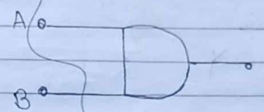
$= 1224+0.125$

$= 1224.125$

$\boxed{(4C8.2)_{16} \rightarrow (1224.125)_8}$

* Logic gate *

AND function



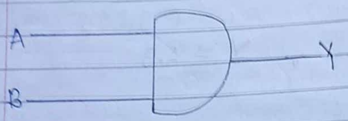
$Y = A.B$

Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Logic gate

Basic gate	Universal gate	derived gate
eg. AND, OR, gate	eg. NAND, NOR	Eg. EX-OR, EX-NOR

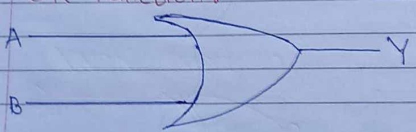
AND Function:-



Input		output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$Y = A \times B$
$= A \cdot B$
$\boxed{Y = AB.}$

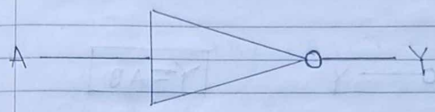
OR function:-



Input		output
A	B	Y
0		
0	0	0
0	1	1
1	0	1
1	1	1

$\boxed{Y = A + B}$

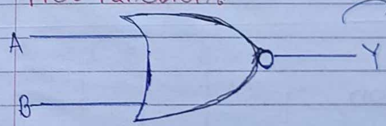
NOT function



Input	output
A	Y
0	1
1	0

$\boxed{Y = \overline{A}}$

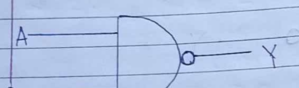
Nor function:-



Input		output
A	B	Y
0	0	1
0	1	1
1	0	0
1	1	0

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

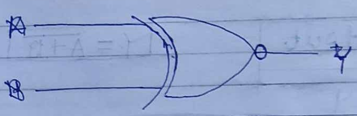
NAND Function :-



$\boxed{Y = \overline{AB}}$

Input		output
A	B	Y
0	0	1
0	1	1
1	0	0
1	1	0

XNOR function



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

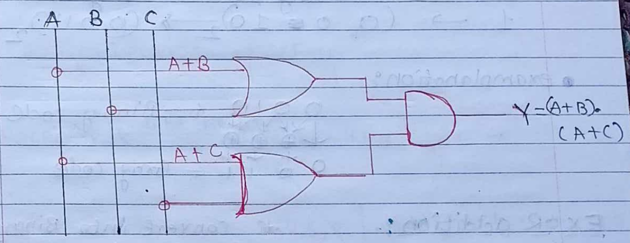
Truth table

Input		output
A	B	$Y = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

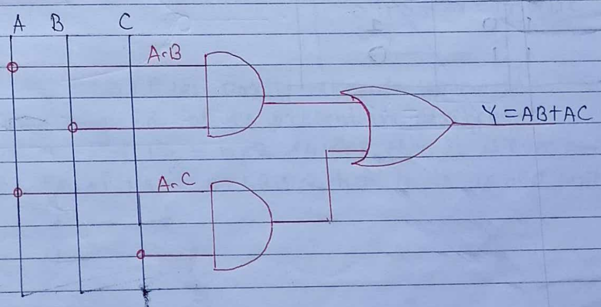
* Problem *

Draw the combination circuit using Basic Gates

① $Y = (A+B).(A+C)$



② $Y = AB + AC$



~~Hammin~~

* Gray Code :-

Binary No. → Gray code.

$1 \rightarrow (0\ 0\ 1\ 0)_2 \longrightarrow (0\ 0\ 1\ 1)_2$

• examplanation :-

0 0 1 0 ← Binary code

$\oplus$ $\oplus$ $\oplus$

0 0 1 1 ← Gray code

Ex OR addition :-

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

* convert into Binary code
* first written 1st no as it is or same in Gray code
* Add with each other

* Error Detecting AND CORRECTING CODES *

↓ Parity checking

↓ Hamming code

1) Parity checking

→ Even

→ odd

| Receiver | ← 4 bit data | Transmitter |

Even Parity :- n = 2, 4, 6, 8

$\underbrace{0\,0\,1\,0}_{\text{original signal}}\ \underset{\text{parity bit}}{1}$

odd parity :- n = 1, 3, 5, 7

$\underbrace{0\,0\,1\,0}_{\text{original signal}}\ 0$ → parity bit

$\underbrace{0\,1\,1\,0}_{1}\ 1$ → parity bit

* ~~bit~~ ODD / Even parity depends upon the how many no. of 1's present in parity boxes
* if no. of 1's = n = 2, 4, 6, 8, then it is even pari[ty]
* If no. of 1's is n = 1, 3, 5, 7, then it is is odd parit[y]

It is the simplest technique for detecting and correcting error. The MSB of an 8-bit word is used as the parity bit and the remaining 7 bits are used as data or message bits. The parity of 8 bits transmitted word can be either even parity or odd parity

MSB							LSB
P	d6	d5	d4	d3	d2	d1	d0

Even.

* Binary Arithmetic

① Binary Addition :-

• Rule :- 1 + 1 = 0 and carry = 1

other addition like normal math

Eg. $(00101100)_2 + (10110101)_2 = (\quad)_2$

Ans :-

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

② Binary Subtraction :-

• Rule :- 0 − 1 = 1 and carry = 1
and 1 − 1 = 0
other subtraction like normal math
and
10 digit consider as 2 in decimal form

Eg $(10101.100)_2 - (00111.000)_2$

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

It is the simplest technique for detecting and correcting error. The MSB of an 8-bit word is used as the parity bit and the remaining 7 bits are used as data or message bits. The parity of 8 bits transmitted word can be either even parity or odd parity

MSB | LSB

P	d_6	d_5	d_4	d_3	d_2	d_1	d_0

Even.

* Binary Arithmetic

① Binary Addition :-

- Rule :- 1 + 1 = 0 and carry = 1

  other addition like normal math

Eg. $(00101100)_2 + (10110101)_2 = (\quad)_2$

Ans :-

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

② Binary Subtraction :-

- Rule :- 0 − 1 = 1 and carry = 1
  and 1 − 1 = 0
  other subtraction like normal math
  and
  10 digit consider as 2 in decimal form

Eg $(10101.100)_2 - (00111.000)_2$

```
  ~~1~~0 0 1 0 1 . 1 0 0
-   0 0 1 1 1 . 0 0 0
---------------------
    0 1 1 1 0 . 1 0 0
```

* Represation of Signed number *

1) Sign magnitude form

2) Complement form

a) 1's complement:-

The 1's complement of a binary number is obtained by complemeting every bit of number i.e. a 0 is changed to 1 and vice-versa.

Rule :- 0 $\xrightarrow{\text{changed}}$ 1

1 $\xrightarrow{\text{changed}}$ 0

e.g. Given number $\Rightarrow$ $(0\,1\,0\,1\,1\,0\,0)_2$

1's complement of the number $\Rightarrow$ $(1\,0\,1\,0\,0\,1\,1)$

2) 2's complement

Rule
- * Take 1's complement
- * Add 1

eg Given no. 1 0 1 0 1 0 1 0

1's complement of no. 0 1 0 1 0 1 0 1

+ 1

0 1 0 1 0 1 1 0

* Law of Boolean Algebra

1) Complementation Law

$A = 0$ then $\overline{A} = 1$

$A = 1$, then $\overline{A} = 0$

2) Commutative Law

(i) $A \cdot B = B \cdot A$ (ii) $A + B = B + A$

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

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

3) Associative Laws

(i) $A + (B + C) = (A + B) + C$ (ii) $A \cdot (BC) = (AB) \cdot C$

4) Distibutive Laws

(i) $A(B + C) = AB + AC$

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

5) AND Law :-

(i) $A \cdot 0 = 0$ (ii) $A \cdot 1 = A$

(iii) $A \cdot A = A$ (iv) $A \cdot \overline{A} = 0$

6) OR Law

(i) $A + 0 = A$ (ii) $A + 1 = 1$

(iii) $A + A = A$ (iv) $A + \overline{A} = 1$

7) Inversion Law

$$\overline{\overline{A}} = A$$

Problem based on Laws

(1) $A\overline{B} + \overline{A}B + AB + \overline{A}\,\overline{B} = 1$

→

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

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

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

$\overline{B} + B = 1$

(2) $A + AB = A$

$= A(1+B)$

$= A \cdot 1 = A$

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

$= AA + AC + BA + BC$

$= A^2 + AC + AB^2 + BC$

$= A\underbrace{(1+C)}_{1} + AB + BC$

$= A + AB + BC$

$= A(1+B) + BC$

$= A + BC$

(4) $A + AB + AB = A + B$

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

$= A + B$

(5) $\overline{AB} = \overline{A} + \overline{B}$ ← 1st De Morgan's Theorem

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

2nd De Morgan's theorems :-

(6) $\overline{A+B} = \overline{A} \cdot \overline{B}$

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

* Codes *

i) Binary codes :-

Encoding the data:-
When the data or information is coded, the number, letter or words are represented by a specific group of symbol. The process of coding these symbol is called as encoding the data.

code:- The group of symbols is called as code.

Binary code :-
The digital data is represented, stored and transmitted in the form of a strem (group) of binary 0's and 1's. This group of binary bits is also called as binary code.

Classification of codes

→ 1. **Weighted code :-**
Each position has fixed weight

OR

In weighted codes, each digit position of the number represents a specific weight.

eg. 8 4 2 1 ⇒ Weight
↑ ↑ ↑↑
n_3 n_2 n_1 n_0 ⇒ position

2. **NON-Weight code :-**

3) Reflective code :-

It is self complimenting code.

In that code 9 is compliment of 0

9	is compliment of	0
8	-II-	1
7	-II-	2
6	-II-	3
5	-II-	4

Decimal	2421
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111

compliment

9 is compliment of 0

4) Sequential code :-

Next data will increament by 1 or difference between suceesive code is 1

e.g 0 → 0000 diff = 0 - 1 = -1

1 → 0001

5. Alphanumeric codes

(ASCII) (EBCDIC)

6. Error detecting and correcting codes

BCD code :-

① In this code, Each decimal digit reprented by 4 bits binary code

② In decimal, digit are ranging from 0 to 9, Which we will represent in 4 bit binary by BCD code

	Decimal		8421	
e.g	0	→	0000	← BCD code
	1	→	0001	

Exces 3 code :-

Decimal NO → 8421 BCD code $\xrightarrow{\text{Add } 0011}$ Excess3

e.g 4 → 0100 → 0100 + 0011 = (0111) Excess-3

DATE
PAGE

* ERROR DETECTING AND CORRECTING CODES.

(i) Parity bit are the extra bit/bits added to the data being transmitted for detecting the error in data transmission.

(ii) The MSB of an 8-bits word is used as the parity bit and the remaining 7 bits used as data or message bits. The parity of 8-bits transmmitted word can be either even parity or odd parity

MSB							LSB
P	d6	d5	d4	d3	d2	d1	d0

Parity bit ← 7 data bits →

Even Parity :- Even parity means the numbers of 1's in the given word including the parity bits should be even (2, 4, 6 ....)

Odd Parity :- odd parity means the number of 1's in the given word including ~~of 1's~~ the parity bit should be odd (1, 3, 5)

DATE
PAGE

Use of Parity Bit :-

(i) The parity bit can be set 0 and 1 depending on the type of the parity required

(ii) For even parity, this bit is set to 1 or 0 such that no. of "1 bits" in the entire word is even. shown in fig (a).

Parity → | 0 | 1 0 0 1 0 1 1 | Parity ↑ | 1 | 1 0 0 1 0 1 1 |

(iii) for odd parity, the bit is set to 1 or 0 ~~which~~ such that the no. of 1's or "1 bits" in the entire word is odd. shown in fig (b).

| 1 | 1 0 0 1 0 1 1 | | 0 | 1 0 0 0 1 1 0 |
↑ Parity   ↑ Parity

How does parity checking detect error

Parity checking at the receiver can detect the presence of an error if the parity of reciver signal is different from the expected parity. That means, if it is known the parity of the transmitted signal is always going to be "even" and if the recived signal has an odd parity, then the reciver can conclude that receiver signal is not correct. If an error is detected, then the receiver will ignore the received byte and request for retransmission of the same byte to the transmitter.

Transmitted code → P | 0 | 1001011 | ——→ P | 0 | 0001011 | ← Received code with error

* Hamming codes *

(i) It is developed by R.W. HAMMING, it is easy to implement 7 bit hamming code we use mostly

(ii) Data Bit ⇒ 4

Parity Bit ⇒ 3 ⇒ 2^n

$n = 0, 1, 2, 3 \ldots n$

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

D7	P6	D5	P4	D3	P2	P1

$P_1 = D_3\, D_5\, D_7$

$P_2 = D_3\, D_6\, D_7$

$P_4 = D_5\, D_6\, D_7$

7	6	5	4	3	2	1
D_7	D_6	D_5	P_4	D_3	P_2	P_1

$P_1 \rightarrow P_1, D_3, D_5, D_7 \rightarrow (1, 3, 5, 7)$

$P_2 \rightarrow P_2, D_3, D_6, D_7 \rightarrow (2, 3, \quad)$

$P_3 \rightarrow P_4, D_5, D_6, D_7 \rightarrow$

2.

Difference between computer architecture and computer organization :-

Computer Architecture	Computer organization
5) All the high-level design issue are handled by the computer architecture	(6) All the low-level design issue are handled by the computer organiz
(7) Computer architecture deals with high-level design issue	(7) computer architecture deals with low-level design issue.
(8) Actors in computer architecture are hardware part	(8) Actor in computer organization is performan
(9) computer architecture is also called as instruction set architecture.	(9) computer organization is frequently called as micro architecture.