# ECE213: Digital Electronics


Ajmer Singh

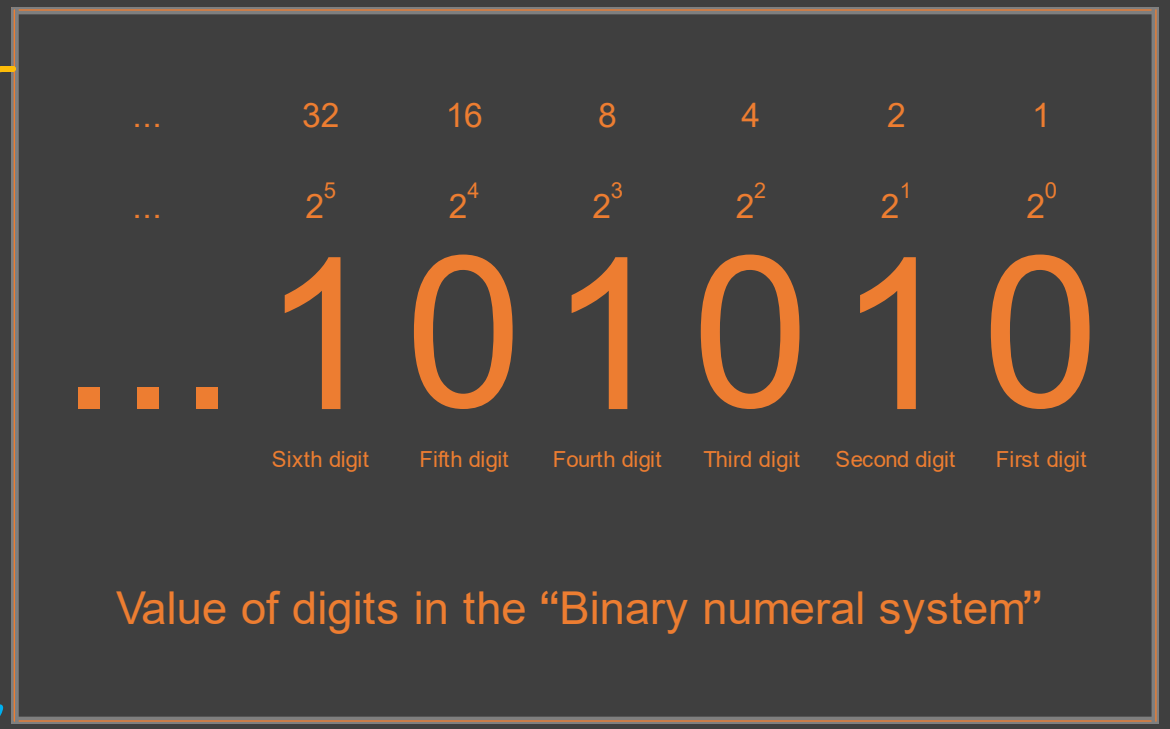
9988921373

ajmer.17381@lpu.co.in


# The Course Contents

## Unit I

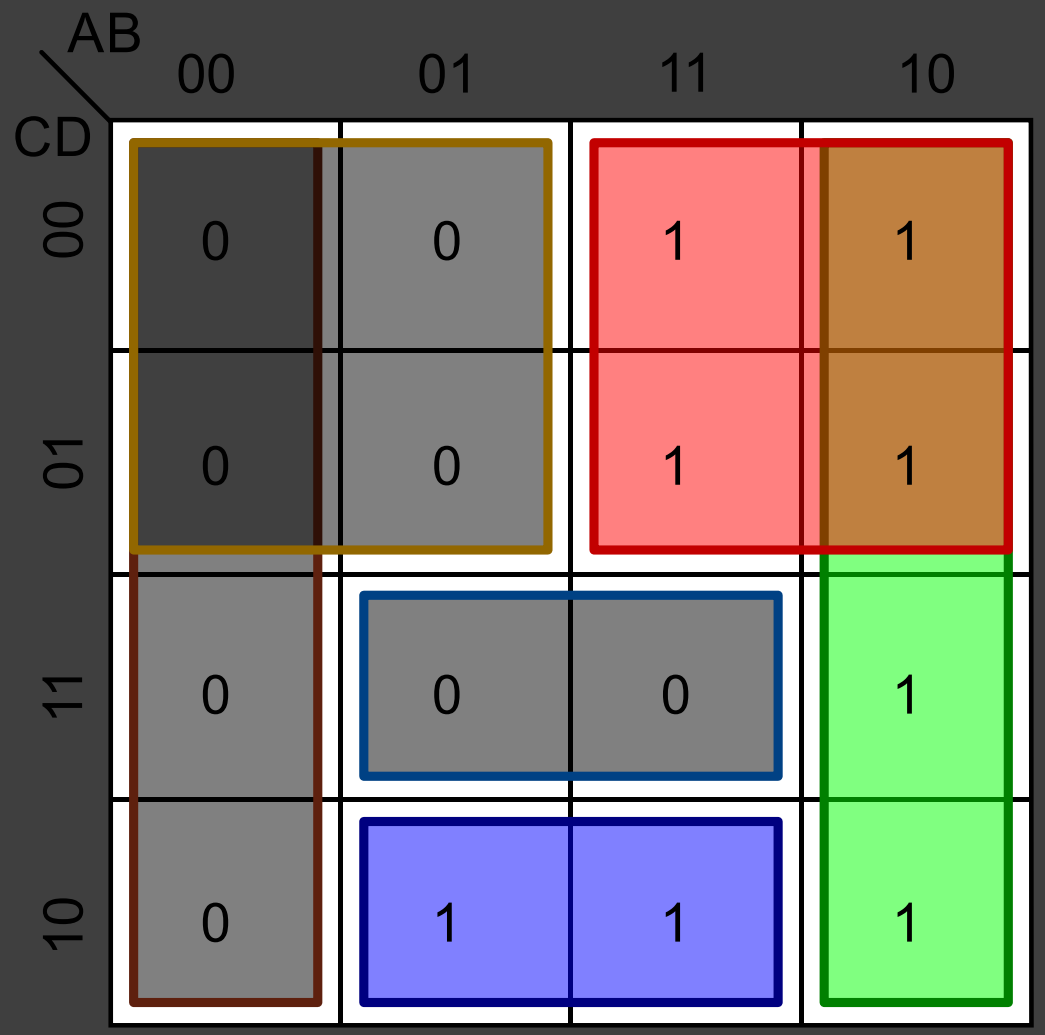
Number Systems : Digital Systems, Data representation and coding, Logic circuits, Implementation of digital systems, Number Systems, Codes- Positional number system, Binary number system, Methods of base conversions, Binary arithmetic, Representation of signed numbers, Fixed numbers, Binary coded decimal codes, Gray codes, Error detection code, Parity check codes, octal number system, Hexadecimal number system, Error correction code, Hamming code, Octal arithmetic, Hexadecimal arithmetic, Floating point numbers



# The Course Contents

## Unit II

Combinational Logic System : Truth table, Basic logic operation, Boolean Algebra, Basic postulates, Standard representation of logic functions -SOP forms, Simplification of switching functions - K-map, Synthesis of combinational logic circuits, Logic gates, Fundamental theorems of Boolean algebra, Standard representation of logic functions POS forms

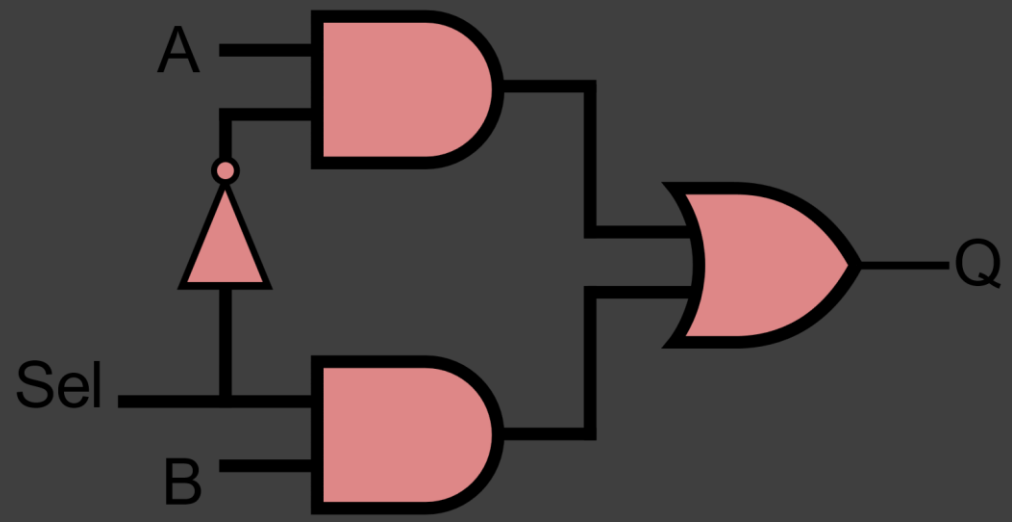


# The Course Contents

## Unit III

**Introduction to Combinational Logic Circuits :** Adders, Subtractors, Comparators, Multiplexers and Demultiplexers, Decoders, Encoders, Parity circuits

**Introduction to Logic Families :** Introduction to different logic families, Structure and operations of TTL, MOS and CMOS logic families

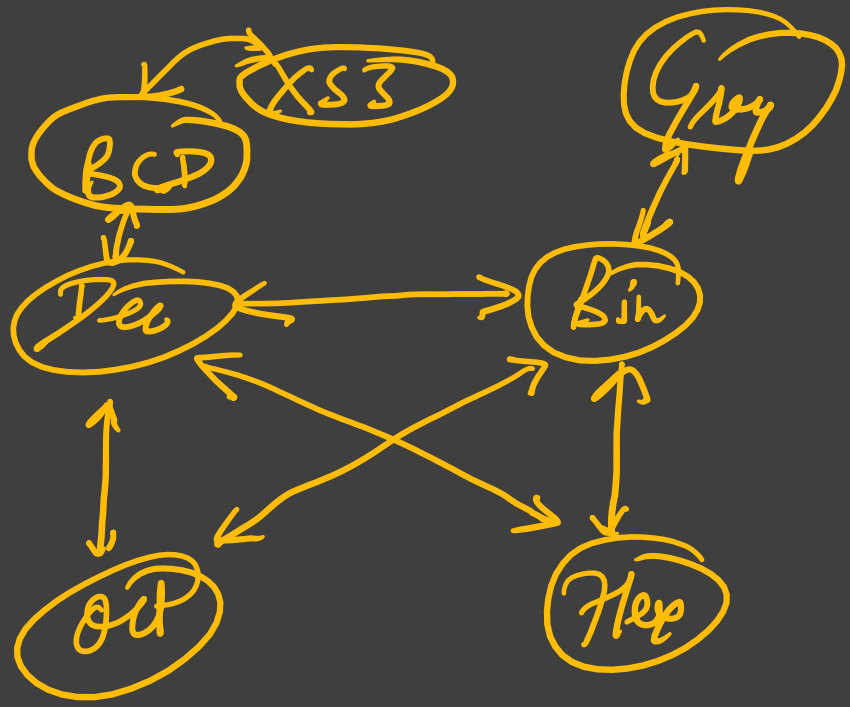


Tutorial

Ex $(AB.0C)_{16} = (\quad)_2 = (\quad)_8 = (\quad)_{10}$

$0(10|0|011.0000|1100)_2$

$(253.030)_8$




1 0 1 0 1 0 1 1 . 0 0 0 0 1 1

128 64 32 16 8 4 2 1 $\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{8}$ $\frac{1}{16}$ $\frac{1}{32}$ $\frac{1}{64}$

$128 + 32 + 8 + 2 + 1 + \frac{1}{32} + \frac{1}{64} = (171.046\ldots)_{10}$

# Tutorial

Ex $(25)_{10} = (100)_b$

Find b.

Sol.

$25 = 1\,0\,0$

$b^2\ b^1\ b^0$

$25 = b^2$

$b = \pm 5$

$b = +5$

Ex $(51)_{10} = (123)_b$

$51 = b^2 + 2b + 3$

$b^2 + 2b - 48 = 0$

$b = 6, -8$

$b = 6$

$(51)_{10} = (123)_6$

# Tutorial

Ex IEEE 754

$(-27.125)_{10}$

| | | |
|---|---|---|
| 2 \| 27 | | |
| 2 \| 13 | 1 | |
| 2 \| 6 | 1 | |
| 2 \| 3 | 0 | |
| | 1 | 1 |

$0.125 * 2 = .250 \quad 0$

$.25 * 2 = 0.5 \quad 0$

$.5 * 2 = 1.0 \quad 1$

$(11011.001)_2 = (27.125)_{10}$

$(4)_{10} = (100)_2$

Normalization

$11011.001 \times 2^0$

$1.1011001 \times 2^4$

```
  1111111
    100
-------
 10000011
```

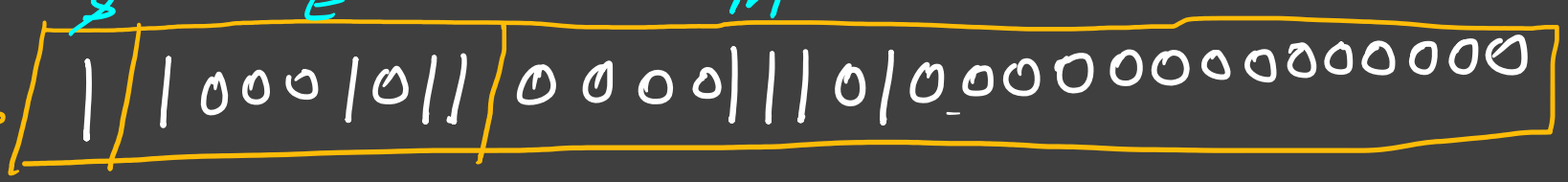
Sign | 8-bit Exponent | 23-bit Mantissa

| 1 | 10000011 | 10110010000000000000000 |
|---|---|---|

with 127 as bias

Tutorial

Ex The Given IEEE 754 representation find the no




-ve

$$N = (-1)^S \, 1.M \times 2^{E-127}$$

$$= (-1)^1 \, 1.00001110100000000000000 \times 2^{12}$$

$$= (-1)^1 \, 1000011101000000000000$$

64 32 16 8 4 2 1

128 1024 2048 4096

$$(-1)\, 4096 + 128 + 64 + 32 + 8$$

$$= -4328$$

1000 0 1 1

1 1 1 1 1 1 1

0 0 0 0 1 1 0 0

12

Tutorial

## ★ Universal Gate — Implement the logic function using NAND Gate

We know $Y_{NAND} = \overline{AB}$ — ①

### ① NOT using NAND

$Y_{NOT} = \overline{A}$

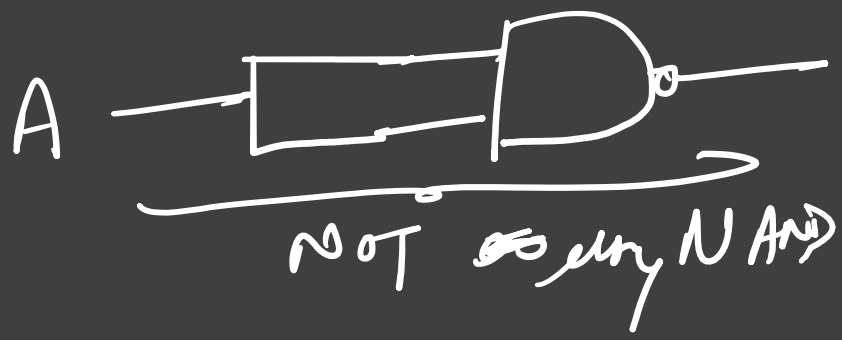
$Y_{NOT} = \overline{AA}$

### ② AND using NAND

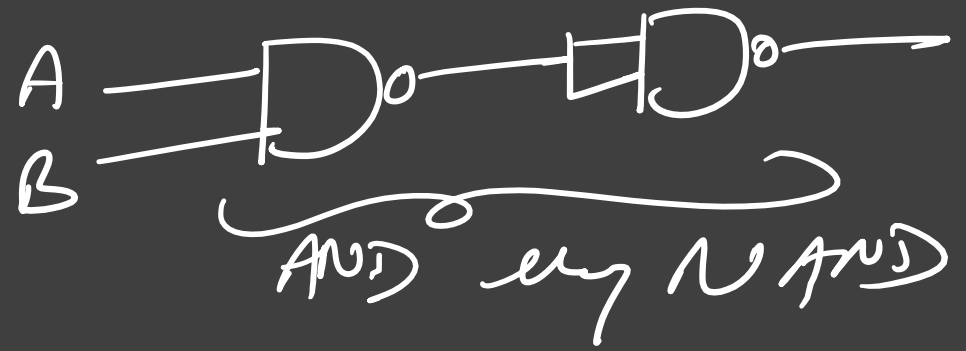
$Y_{AND} = AB$

$= \overline{\overline{AB}}$

$= \overline{\overline{AB}}$




NOT using NAND.



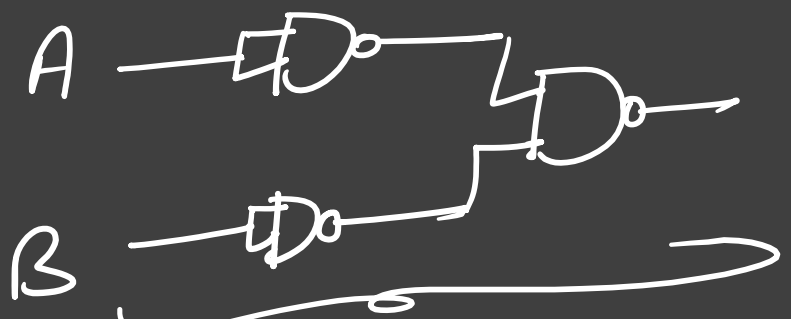

AND using NAND

Tutorial

$Y_{NAND} = \overline{AB}$

## (3) OR Gate using NAND

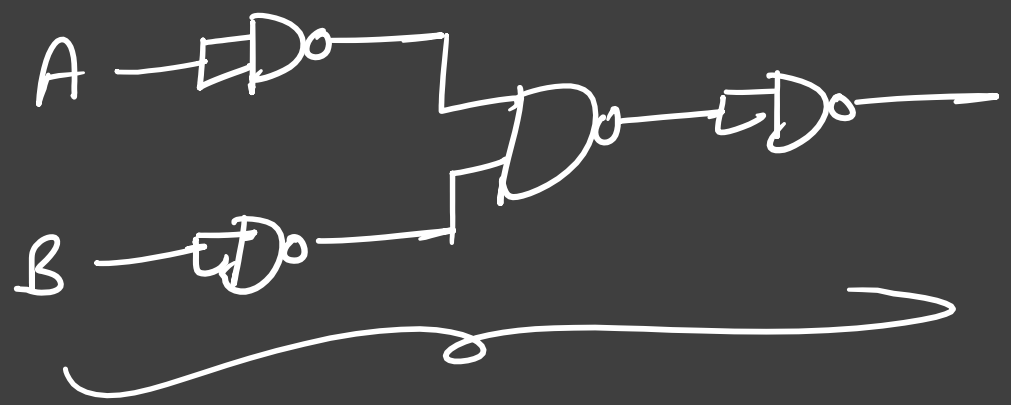
$$Y_{OR} = A + B$$
$$= \overline{\overline{A+B}}$$
$$= \overline{\overline{A}\,\overline{B}}$$




OR using NAND

## (4) NOR Gate using NAND

$$Y_{NOR} = \overline{A+B}$$
$$= \overline{\overline{\overline{A}\,\overline{B}}}$$



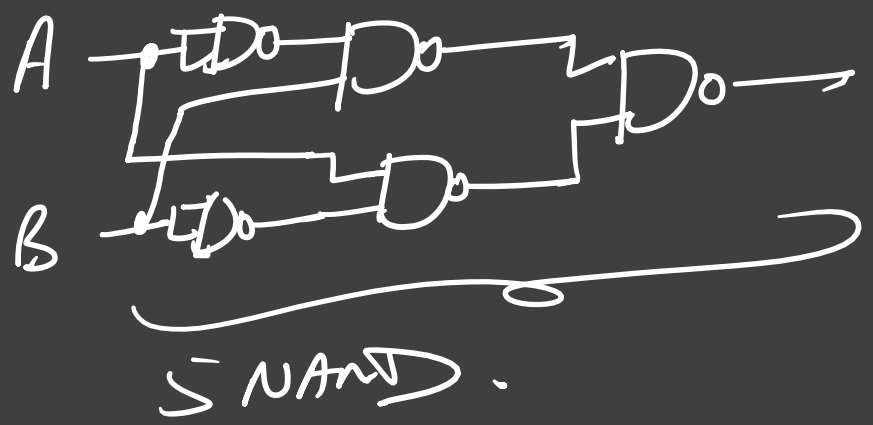
Tutorial

(5) XOR Gate using NAND

$$Y_{XOR} = A\bar{B} + \bar{A}B$$

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

$$= \overline{\overline{A\bar{B}} \cdot \overline{\bar{A}B}}$$




$$Y_{NAND} = \overline{AB}$$

H.W: Find How to represent XOR Gate using only 4 NAND Gate