UNIT III

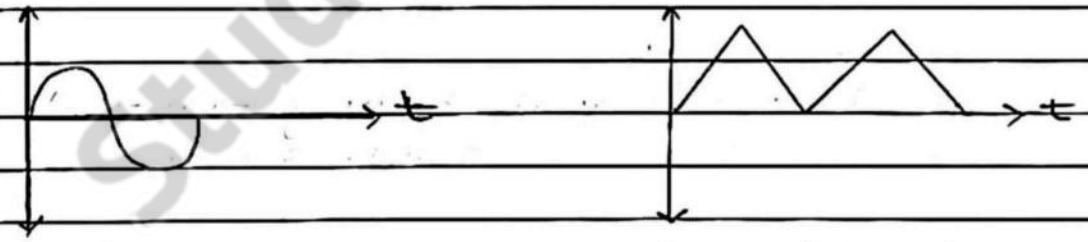
Number System and Logic Gates

* Signals :-

- Signal is a physical quantity which contains some information and which is a function of one or more independent variables.
- The signals can be of two types:
  1) Analog signals
  2) Digital signals

*Analog Signals :

- An analog signal is defined as the signal having continuous values. They can have infinite number of different values.
- Eg. of analog signal. sine wave, triangular wave



Sine wave triangular wave

*Digital Signals :

- A digital signal is defined as the signal which has only a finite number of distinct values.
- Digital signals are not continuous signal.

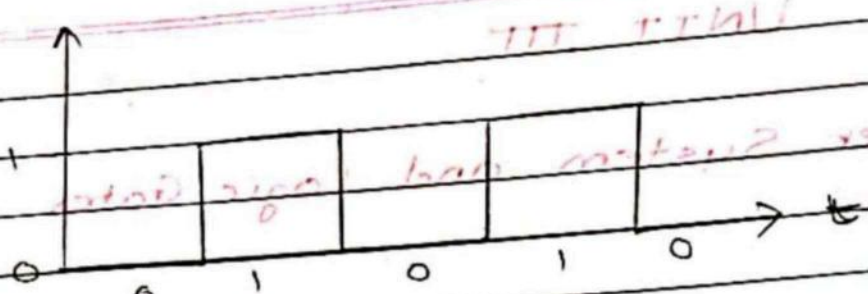


Fig. Binary signal (Digital signal)

Type of digital signal	Number of distinct values
Binary	2
Octal	8
Hexadecimal	16

Binary signal :-

- If a digital signal has only two distinct values, i.e. 0 and 1 then it is called as a binary signal.

Octal signal :-

- A digital signal having eight distinct value is called as an octal signal.

Hexadecimal signal :

- A digital signal having sixteen distinct values is called as the hexadecimal number.

* Number System :-

I) Decimal Number System

- In decimal number system we can express any decimal in units, tens, hundreds, thousands and so on.
- When we write a decimal number 5678.9, we ~~can~~ know it can be represented as

$5000 + 600 + 70 + 8 + 0.9 = 5678.9$

- The decimal number 5678.9 can also be written as $(5678.9)_{10}$, where the 10 indicate the radix or base.

10^3	10^2	10^1	10^0		10^{-1}
5	6	7	8	.	9
5×10^3	6×10^2	7×10^1	8×10^0	.	9×10^{-1}
MSD					LSD

II) Binary Number System

- Binary system with its two digits is a base-two system.
- The two binary digits (bits) are 1 and 0.
- In binary system, weight is expressed as a power of 2.

eg.

N =	1	1	0	1	.	1	0	1
	1×2^3	1×2^2	0×2^1	1×2^0	.	1×2^{-1}	0×2^{-2}	1×2^{-3}

$N = (13.625)_{10}$

III) Octal Number System:

- The octal number system uses first eight digits of decimal number system: 0, 1, 2, 3, 4, 5, 6 and 7. As it uses 8 digits, its base 8.

eg.

$N = 5\ 6\ 3\ 2\ .\ 4\ 7\ 1$

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

$N = (2970.611328)_{10}$

IV Hexadecimal Number System

- The hexadecimal number system has a base of 16 having 16 characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 A, B, C, D, E and F

Decimal	Binary	octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

eg. 3FD.84

N = 3 F D . 8 4

3×16^2 $F\times16^1$ $D\times16^0$. 8×16^{-1} 4×16^{-2}

$N = 3\times16^2+15\times16^1+13\times16^0+8\times16^{-1}+4\times16^{-2}$

$= (1021.515625)_{10}$

* Conversion between Binary, Octal, Decimal and Hexadecimal

I) Decimal to Binary

- Successive division for integer part conversion

Steps :-

a) Divide the integer part of given decimal number by the base and note down the remainder.

b) Continue to divide the quotient by the base (r) until there is nothing left, noting the remainders from each step.

c) List the remainder values in reverse order from the bottom to top to find the equivalent

eg. $(105)_{10}$

2	105	
2	52	1 LSB
2	26	0
2	13	0
2	6	1
2	3	0
2	1	1
	0	1 MSB

$(105)_{10} = (1101001)_2$

- Successive multiplication for Fractional part conversion

Steps :-

a) Multiply the given Fractional decimal numb by the base (r)

b) Note down the carry generated in this multiplication as (MSD)

c) Multiply only the Fractional number of the product in step 2 by the base, and note down the carry as the next bit to MSD.

d) Repeat steps 2 & 3 upto the end. The last carry will represent the LSD of equivalen i.e. converted number.

eg. $(0.42)_{10}$

$0.42 \times 2 = 0.84 \rightarrow 0$ MSB

$0.84 \times 2 = 1.68 \rightarrow 1$

$0.68 \times 2 = 1.36 \rightarrow 1$

$0.36 \times 2 = 0.72 \rightarrow 0$

$0.72 \times 2 = 1.44 \rightarrow 1$ LSB

~~0.04~~ $(0.42)_{10} = (0.01101)_2$

Binary to Decimal

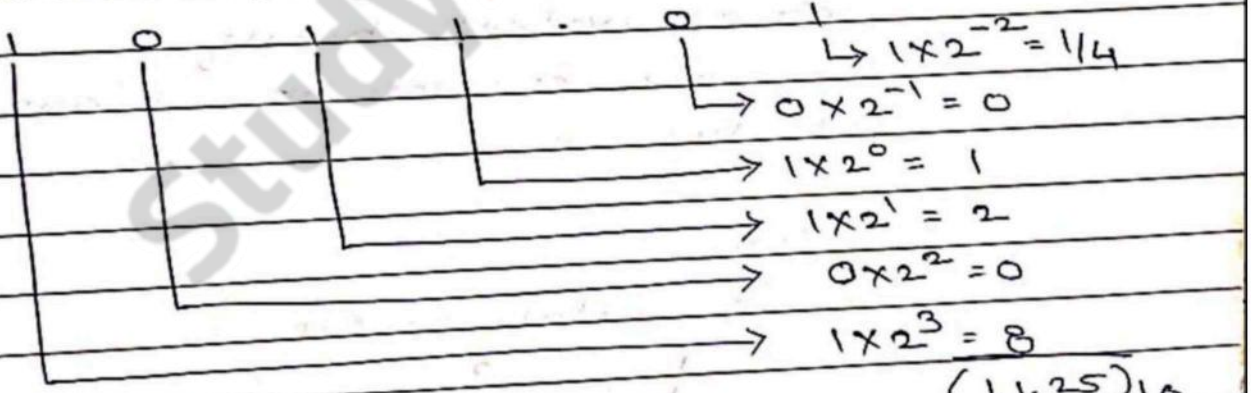
Step 1 : Note down the given number.

Step 2 - Write down the weights corresponding to different positions.

Step 3 - Multiply each digit in the given number with the corresponding weight to obtain product numbers.

Step 4 - Add all the product numbers to get the decimal equivalent.

e.g. (1011.01)



$(1011.01)_2 = (11.25)_{10}$

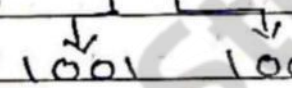
* Binary coded Decimal (BCD):-
- In this code each decimal digit is represent by a 4-bit binary number.
- Thus BCD is a way to express each of the decimal digits with a binary code.

Conversion From Decimal to BCD:-
- The decimal digits 0 to 9 are converted into a BCD, exactly in the same way as binary.

Decimal	0	1	2	3	4
BCD	0000	0001	0010	0011	0100

Decimal	5	6	7	8	9
BCD	0101	0110	0111	1000	1001

e.g. 98



$(98)_{10} = (1001\ 1000)$

Smallest BCD number :- 0000
largest BCD number - 1001

Advantages of BCD codes:-
- It is very similar to decimal system
- We need to remeber binary equivalents of decimal numbers 0 to 9 only.

Disadvantages :-

- The addition and subtraction of BCD have different rules.
- The BCD arithmetic is little more complicated.
- BCD needs more number of bits than binary to represent the same decimal number. So BCD is less efficient than binary.

* Binary Addition

	A	B	Addition(S)	carry
case1	0	0	= 0	0
case 2	0	1	= 1	0
case 3	1	0	= 1	0
case 4	1	1	= 0	1

e.g. $(10111)_2$ & $(11001)_2$

```
A =   1  0  1  1  1
B = + 1  1  0  0  1
      1  1  1  1  1   ← carry.
  1  1  0  0  0  0
```

* Binary Subtraction.

case I :- Digit A > Digit B

let $A = (5)_{10}$ & $B = (3)_{10}$

Then $(A - B) = (5)_{10} - (3)_{10}$

$= (2)_{10}$.

Case II :- Digit A < Digit B.

if $A = (3)_{10}$ & $B = (5)_{10}$

we can not perform it directly, we hav

* logic Gates :-

- logic gates are the logic circuits which act as the basic building blocks of any digital system.
- It is an electronic circuit having one or more than one inputs and only one output.
- The two important characteristics of logic gates are : Truth Table and Boolean expressi

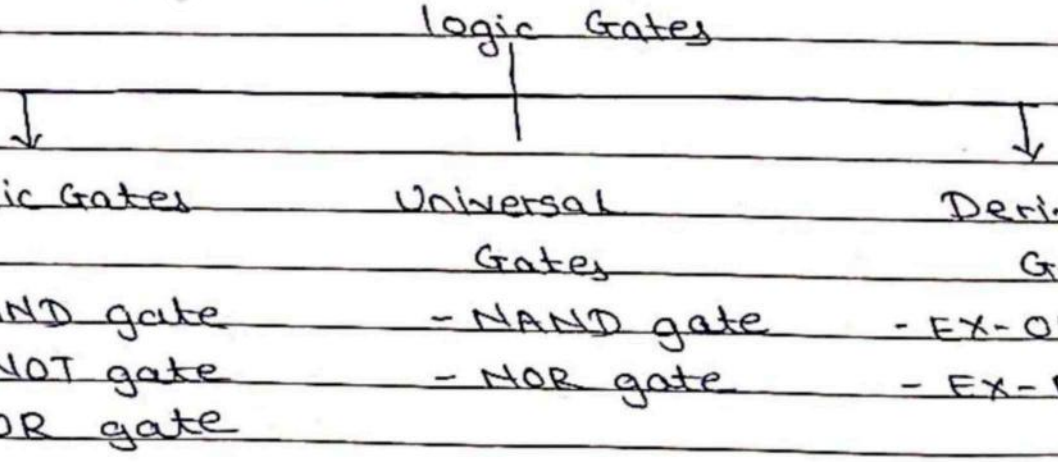
Truth Table :-

- The operation of a logic gate or a logic circuit can be best understood with the help of a table called Truth Table.

Boolean Expression :-

- The relation between the inputs and the outputs of a gate can be expressed mathematically by means of the Boolean expression.

* Classification of logic gates

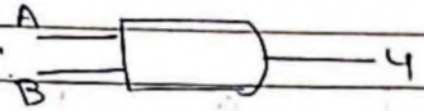


* Basic Gates:

1. AND Gate:

Boolean Expression $Y = A \cdot B$

Symbol:



Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

2. OR Gate

Boolean Expression $Y = A + B$

Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

3. NOT Gate:

Boolean Expression $Y = \bar{A}$

Truth Table

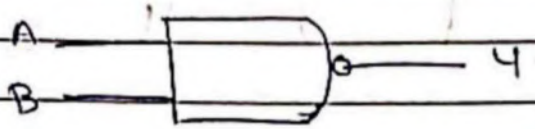
A	Y
0	1
1	0

* Universal Gate :-

1. NAND :- NOT of AND Gate

Boolean Expression :- $Y = \overline{A.B}$

Symbol



IC = 7400

Truth Table

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

2. NOR :- NOT of OR Gate

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

Symbol

Truth Table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

* Derived Gates

1. EX-OR Gate.

Boolean Expression $Y = A \oplus B$

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

Symbol

IC - 7486

Truth Table:

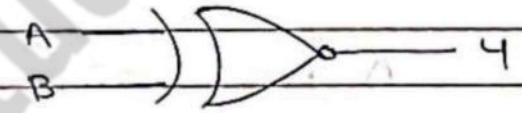
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

2. EX - NOR Gate:

Boolean Expression $Y = A \odot B$ or $A \oplus B$

or $Y = A \cdot B + \bar{A}\bar{B}$

Symbol



Truth Table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

IC → 7426

* Basic Gates
- NOT → 7404
- AND → 7408
- OR → 7432

Universal Gates
- NAND → 7400
- NOR → 7402

Derived Gates
- EX-OR → 7486
- EX-NOR → 74266

* Boolean Algebra :-

Name	AND Form	OR Form
1. AND/OR laws	$1 \cdot A = A$	$1 + A = 1$
	$0 \cdot A = 0$	$0 + A = A$
	$A \cdot A = A$	$A + A = A$
	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$
2. Inversion law	$A = \bar{\bar{A}}$	
3. Commutative laws	$AB = BA$	$A+B = B+A$
4. Associative law	$(AB)C = A(BC)$	$(A+B)+C = A+(B+$
5. Distributive law	$A+(B+C) = (A+B)(A+C)$	$A(B+C) = AB+AC$

6. Absorption law $A(A+B) = A$ $A+(AB) = A$

7. De-Morgan's law $\overline{AB} = \overline{A}+\overline{B}$ $\overline{A+B} = \overline{A}\cdot\overline{B}$

* De-Morgan's law

1. $\overline{AB} = \overline{A}+\overline{B}$

NAND = Bubbled OR

This theorem states that the complement of product of two variables is equal to addition of complements of them.

Proof :-

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

$\therefore \overline{A\cdot B} = \overline{A}+\overline{B}$

2. $\overline{A+B} = \overline{A}\cdot\overline{B}$

NOR = Bubbled AND

This theorem states that complement of addition is equals to multiplication of complement

Proof :

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

$\therefore \overline{A+B} = \overline{A} \cdot \overline{B}$

* Digital circuits.

Combinational circuits	Sequential circuits
1. output depends upon present input only.	output depends upon present input and previous outp
2. Do not required memory	Require memory unit to store previous data.
3. Clock pulses are not required.	Clock pulses are required.
eg. Adder, multiplexer, Comparator	eg. Flip-Flops, Counter, timer.

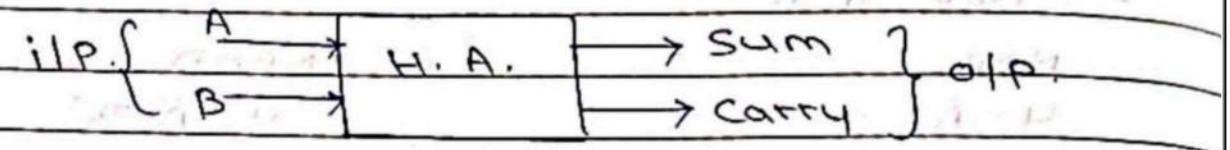
* Combinational circuits :-

Adder

a) Half Adder

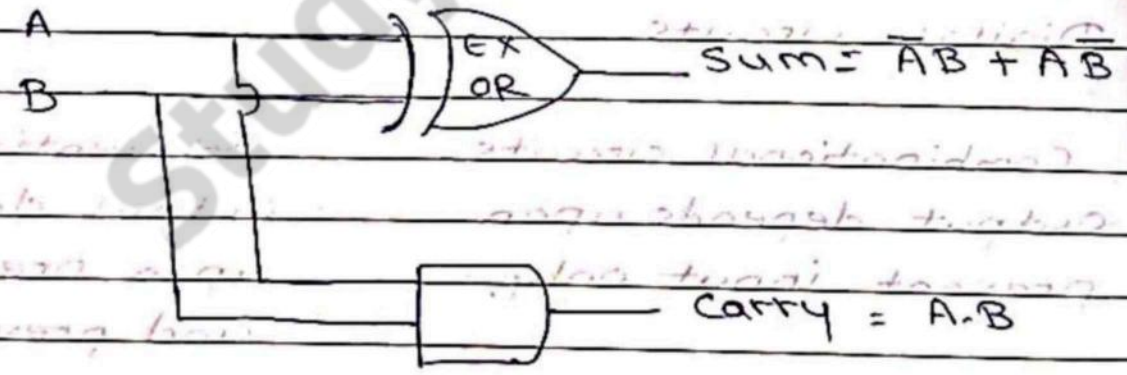
b) Full Adder.

a) Half Adder.



Truth Table

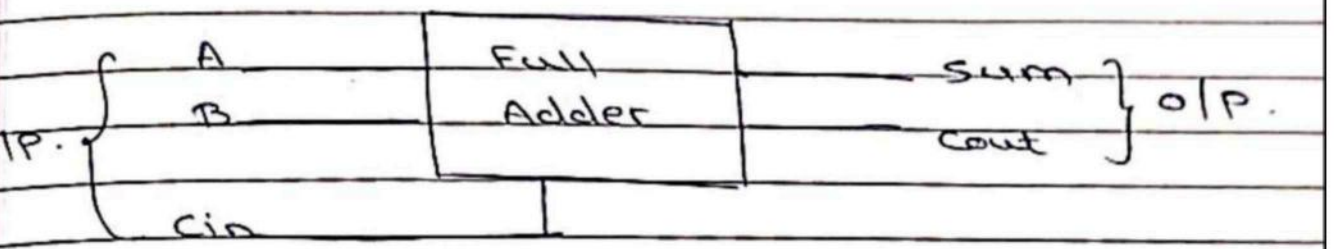
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Disadvantage of half adder :-

÷ Addition of 3 bit not possible using half adder.

b) Full adder :-



Truth table :-

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

expression for Sum

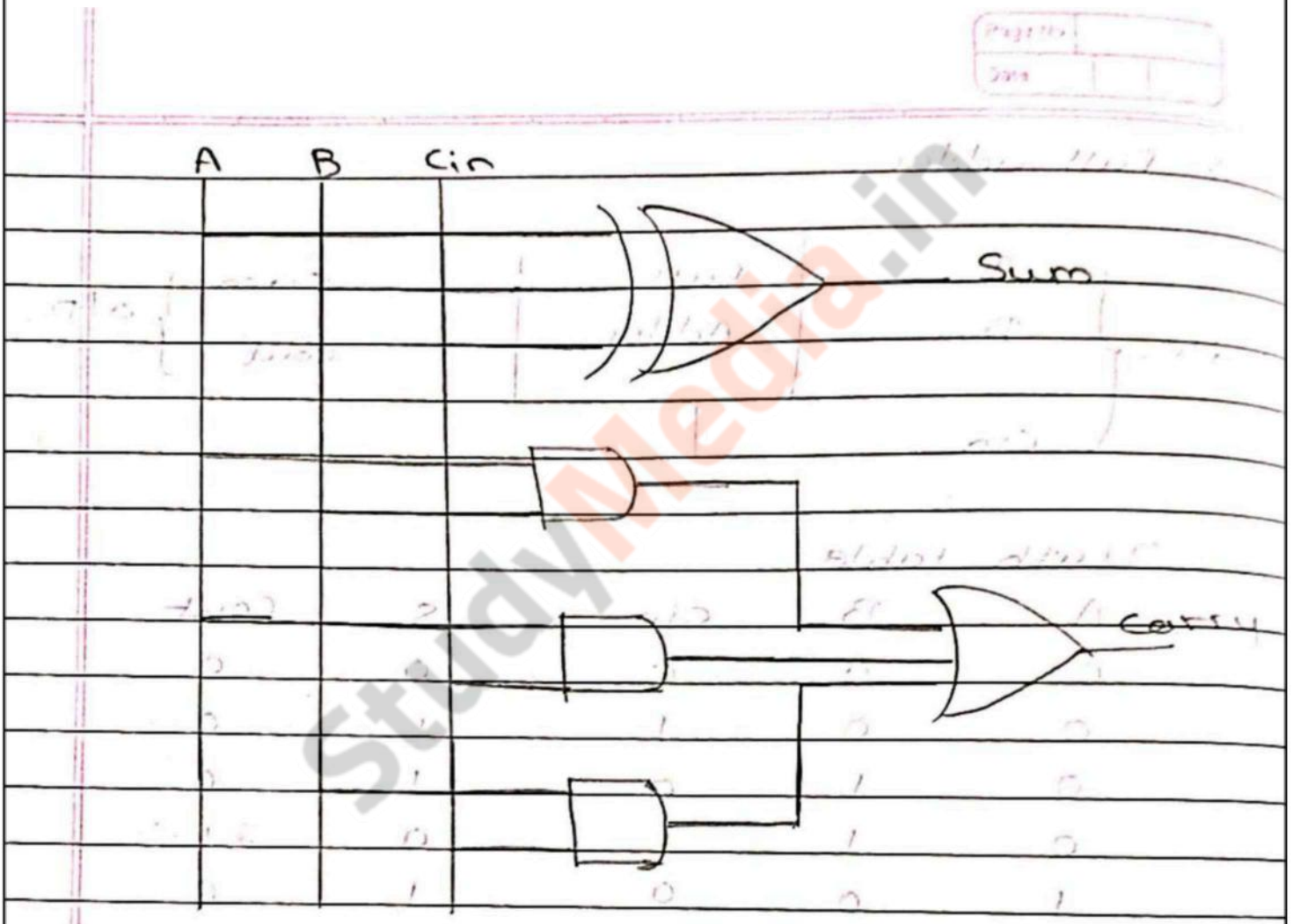
$$S = \bar{A}\bar{B}Cin + \bar{A}B\bar{Cin} + A\bar{B}\bar{Cin} + ABCin$$

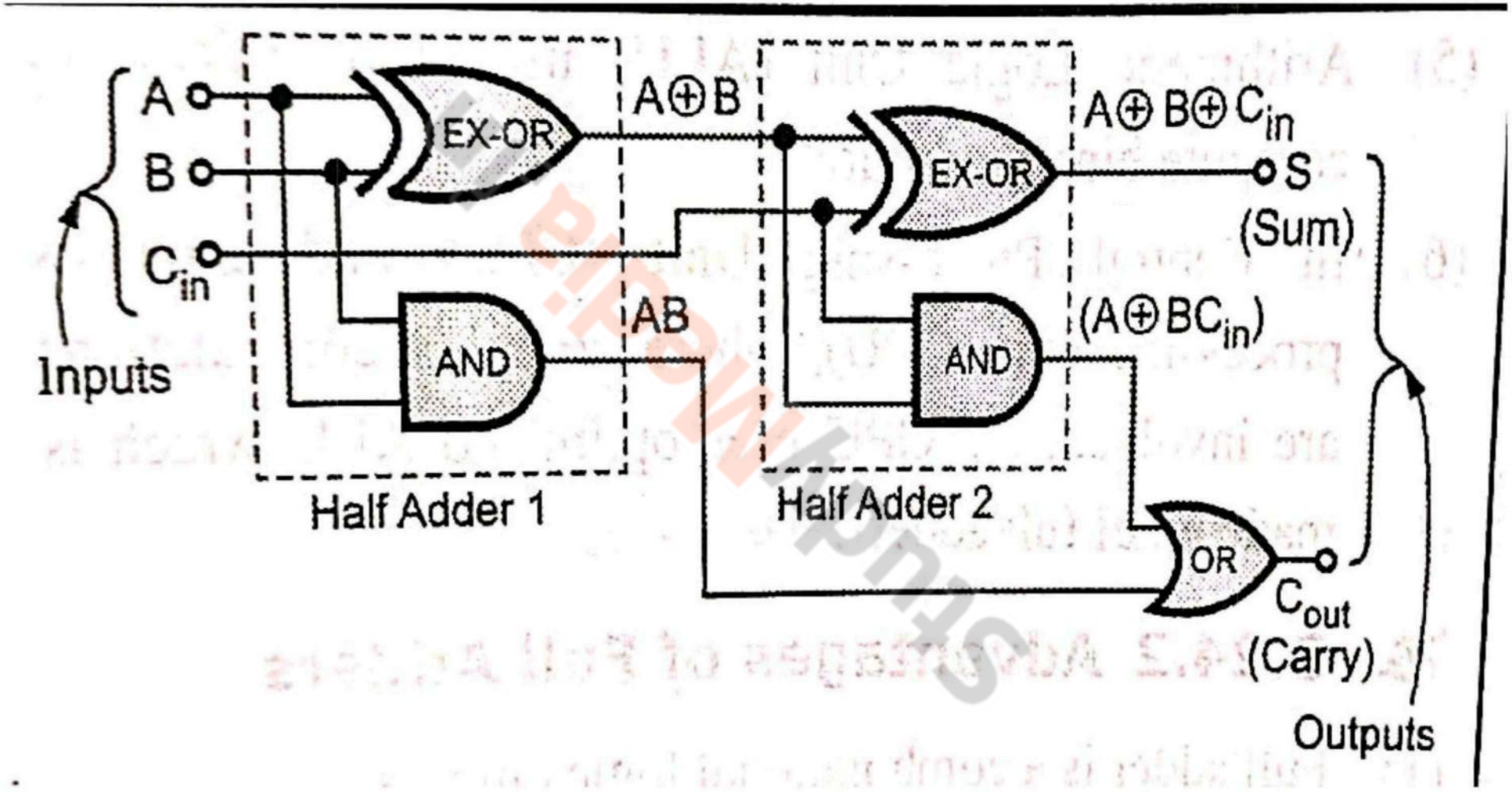
$$= Cin \oplus A \oplus B$$

expression for carry

$$Cout = \bar{A}BCin + A\bar{B}Cin + AB\bar{Cin} + ABCin$$

$$= AB + ACin + BCin.$$





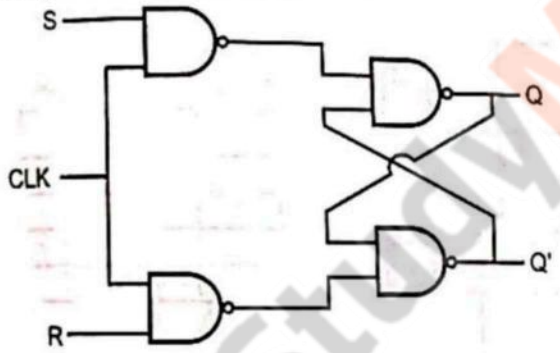
A flip-flop is a sequential digital electronic circuit having two stable states that can be used to store one bit of binary data. Flip-flops are the fundamental building blocks of all memory devices.

3.26.1 Types of Flip-Flops

1. S-R Flip-Flop
2. J-K Flip-Flop
3. D Flip-Flop
4. T Flip-Flop

3.26.2 S-R Flip-Flop

This is the simplest flip-flop circuit. It has a set input (S) and a reset input (R). In this circuit when S is set as active, the output Q would be high and the Q' will be low. If R is set to active then the output Q is low and the Q' is high. Once the outputs are established, the results of the circuit are maintained until S or R get changed, or the power is turned off.



S-R Flip Flop Truth Table

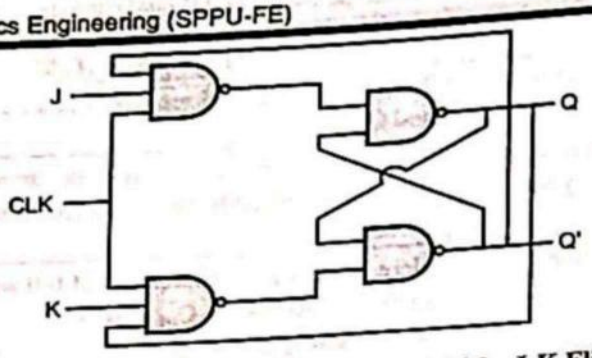
S	R	Q	Q'
0	0	0	No Change
0	1	0	Reset
1	0	1	Set
1	1	x	Forbidden

Fig. 3.26.1 : S-R Flip-Flop

3.26.3 J-K Flip-Flop

(1) Because of the invalid state corresponding to S=R=1 in the SR flip-flop, there is a need of another flip-flop. The J-K flip-flop operates with only positive or negative clock transitions. The operation of the J-K flip-flop is similar to the SR flip-flop. When the input J and K are different then the output Q takes the value of J at the next clock edge.

(2) When J and K both are low then NO change occurs at the output. If both J and K are high, then at the clock edge, the output will toggle from one state to the other.



J-K Flip Flop Truth Table

S	R	Q	Q'
0	0	0	No Change
0	1	0	Reset
1	0	1	Set
1	1	Toggle	Toggle

Fig. 3.26.2 : J-K Flip-Flop

3.26.4 D Flip-Flop

(1) In a D flip-flop, the output can only be changed at positive or negative clock transitions, and when the inputs change at other times, the output will remain unaffected.

(2) The D flip-flops are generally used for shift-registers and counters. The change in output state of D flip-flop depends upon the active transition of clock. The output (Q) is same as input and changes only at active transition of clock.

D Flip Flop Truth Table

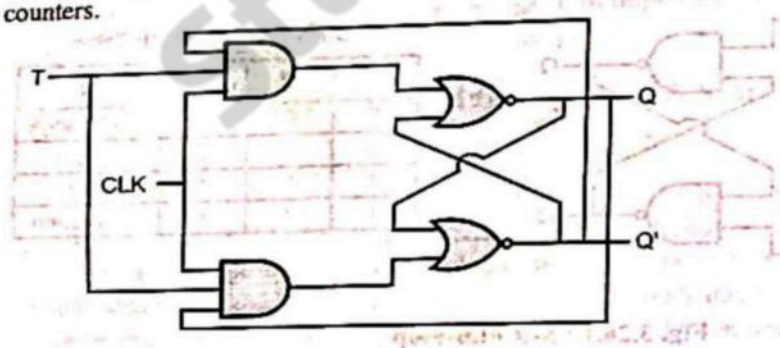
D	Q
0	0
1	1

Fig. 3.26.3 : D Flip-Flop

3.26.5 T Flip-Flop

(1) A T flip-flop (Toggle Flip-flop) is a simplified version of JK flip-flop. The T flop is obtained by connecting the J and K inputs together. The flip-flop has one input terminal and clock input.

(2) These flip-flops are said to be T flip-flops because of their ability to toggle the input state. Toggle flip-flops are mostly used in counters.



T Flip Flop Truth Table

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 3.26.4 : T Flip-Flop

3.26.6 Applications of Flip-Flops

1. Acts as one bit memory
2. In the shift registers
3. Building block of counter (preferred T Flip-flop)
4. In the timers
5. In the clock or frequency dividers

* Microprocessor:

- A microprocessor is defined as a multipurpose, programmable, clock driven, register based electronic device.
- A microprocessor communicates & operates using binary number.
- Each microprocessor has a fixed set of instructions called as machine language.
- Machine language instructions can not be read by human. For reading or operating instructions it forms the assembly language.

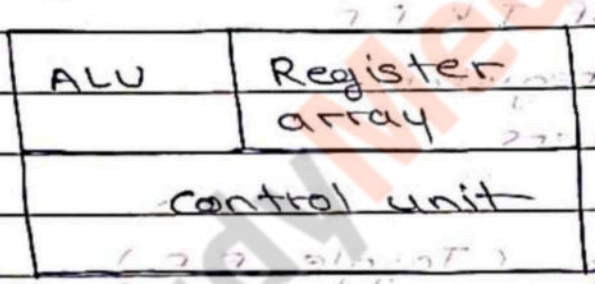


Fig. Block dia. of Microprocessor.

1) ALU:- Arithmatic and logical unit
- It carries out all computations including arithmatic & logical operations on the data.

2) Register Array
- It consists of group of registers.
- Registers are basically used for a temporary

storage of data to be processed.
- All registers are accessible to the user.

3> Control unit :-
- This unit produces all the necessary timing and control signals for all operations
- It controls exchange of data between the microprocessor and memory and I/o devices.

*Available microprocessor
- 8 bit
- 16 bit
- 32 bit
- 64 bit

* Serial Bus :-
- It is used for transferring or carrying from one location to another.
- Types of bus
  a> Address bus
  b> Data bus
  c> Control bus

Applications of microprocessor :-
- CPU of computer
- Traffic control lights.
- In industrial application.

* Microcontroller :-

- It is a device that includes microprocessor
- The memory consists of ROM & RAM.

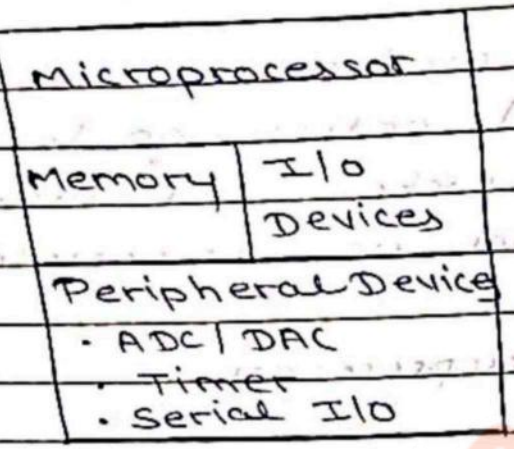


Fig. Block diagram of microcontroller

Features :-

1) Range 4 bit to 32 bit
2) Serial I/O ports
3) RAM for data storage

Applications

1) Remote controls
2) Engine controls
3) Office machines

Microprocessor	Microcontroller
1. It is only a CPU of general purpose computer	• It is microcomputer itself.
2. It does not have inbuilt memory & timer.	• It have inbuilt memory & timer
3. I/o ports are not available	• I/O ports are available.
4. Same memory for program and data	• Separate memory for program and data.
5. less multifunction pins.	• ~~Few instru~~ many multifunction pins on IC.
6. ~~less~~ higher accessing time is required.	• ~~Higher~~ less accessing time is required.
7. Not capable of handling Boolean Functions.	• Capable of handling boolean Functions.
8. Many instructions are needed to read / write data to / From external memory	• Few instructions are needed to read / write data to / From external memory.