I - ASSIGNMENT

(Start Writing From Here)

1. State and prove commutative laws, Associative lawsand distributive law with truthtable.

The Commutative laws are fundamental Principles in Boolean algebra. The commutative laws state that the order of operands doesnot affect the outcome of certain binary operations. In Boolean algebra, the two main commutative operations are the AND operation($\wedge$) are the OR Operation ($\vee$).

Commutative law of AND (conjunction):

$$P \wedge Q = Q \wedge P$$

This law states that the order in which two proposition are combined using the AND operation does not affect the result.

Truth table:-

P	Q	$P \wedge Q$
0	0	0
0	1	0
1	0	0
1	1	1

Commutative law of OR (Disjunction)

$P \vee Q = Q \vee P$

This law states that the order in which two propositions are combined using OR operation does not affect result.

Truth table:-

p	q	p∨q
0	0	0
0	1	1
1	0	1
1	1	1

Associative law:- This law states that the grouping of operands does not affect the outcome of certain binary operations.

The main two Associative operations are the AND operation (∧) and the OR operation (∨).

Associative law of AND:-

The law is $(p \wedge q) \wedge r = p \wedge (q \wedge r)$

Truth table

p	q	r	(p∧q)∧r	p∧(q∧r)
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Associative law of OR :-

This law is $(P \vee q) \vee r = P \vee (q \vee r)$

Truth table :-

P	q	r	$(P \vee q) \vee r$	$P \vee (q \vee r)$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Distributive laws :-

The Distributive law in Boolean algebra describes the relationship between AND($\wedge$) and OR ($\vee$) operations

⇒ There are two forms of distributive law, one for each operation.

Distribution law for AND over OR :-

The law is $P \wedge (q \vee r) = (P \wedge q) \vee (P \wedge r)$

Truth table :-

P	q	r	$q \vee r$	$P \wedge (q \vee r)$	$(P \wedge q) \vee (P \wedge r)$
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Distributive Law for OR over AND:

The law is $p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$

Truth table:

p	q	r	$q \wedge r$	$p \vee (q \wedge r)$	$(p \vee q) \wedge (p \vee r)$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	1

2. Explain about floating-point representation with an Example.

→ Floating-Point representation is a method used to represent real numbers in a way that accommodates a wide range of values, including very large and very small numbers.

→ In a computer System floating-point numbers are typically represented using the IEEE 754 standard.

→ The basic idea to Express a real number as a product of two components: a Mantissa and an Exponent

→ The general form of a floating-point representation can be expressed as

$$(-1)^{Sign} \times Mantissa \times 2^{Exponent}$$

Floating point representation can be represented in two ways single precision - $1.N \times B^{e+127}$

Double precision - $1.N \times B^{e+1023}$

Example:-

$25.5 \quad (11001.1)_2$

11001.1×2^4

$\boxed{1.10011 \times 2^4}$

↓ Mantisa

2	25
2	12 - 1
2	6 - 0
2	3 - 0
	1 - 1

$(11001)_2$

0.5×2

$= 1.0$

Single precision

$1.N \times B^{e+12}$

$1.10011 \times 2^4 = 1.N \times B^{e+127}$

$1.10011 \times 2^4 = 1.10011 \times B^{4+127}$

$= 131$

32 Bits

Sign	E	M
↓ 1	↓ 8 bits	↓ 23 bits

0	10000011	10011000000000
		00000000000

Exponent

2	131
2	65 - 1
2	32 - 1
2	16 - 0
2	8 - 0
2	4 - 0
2	2 - 0
	1 - 0

Double Precision:-

$1.N \times B^{E-1023}$

$1023+4$

$= 1027$

Sign	E	M
1 bit	11 bit	52 bit

2	1027
2	513 - 1
2	256 - 1
2	128 - 0
2	64 - 0
2	32 - 0
2	16 - 0
2	8 - 0
2	4 - 0
2	2 - 0
	1 - 0

0	10000000011	10011000000000000000000000000000000000000000000000
Sign	Exponent	Mantissa

3) What do you mean by decoder? Design a 3-to-8 line decoder and explain it

Decoder:- Decoder is a combinational circuit that convert one type of input into another type. It is a multiple input, multiple output logic circuit.

Decoder is a logic circuit which converts binary information from n input lines to a maximum of 2^n unique output lines with one enable line that is used to

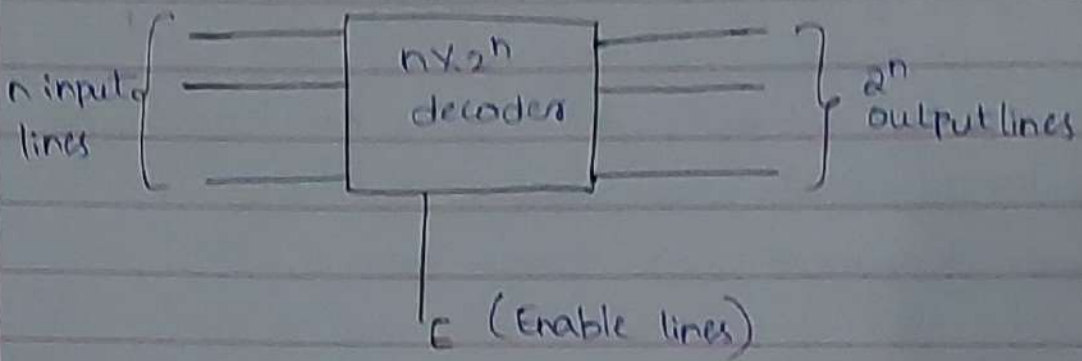
activate or deactivate circuit

The standard form of decoder is $n \times 2^n$ where $(n = 1, 2, 3 \cdots)$

n = Number of input lines

2^n = Number of output lines

Block diagram of decoder

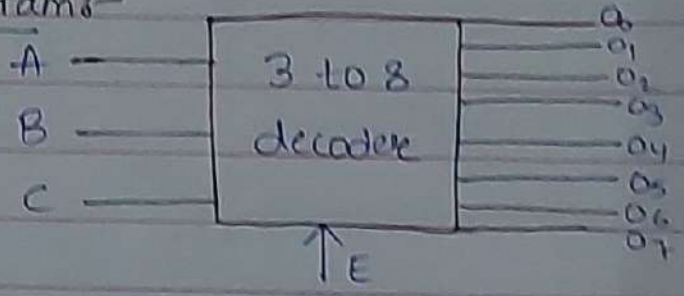


3 to 8 line decoder:-

number of input = 3

number of output = 8

Block diagram:-

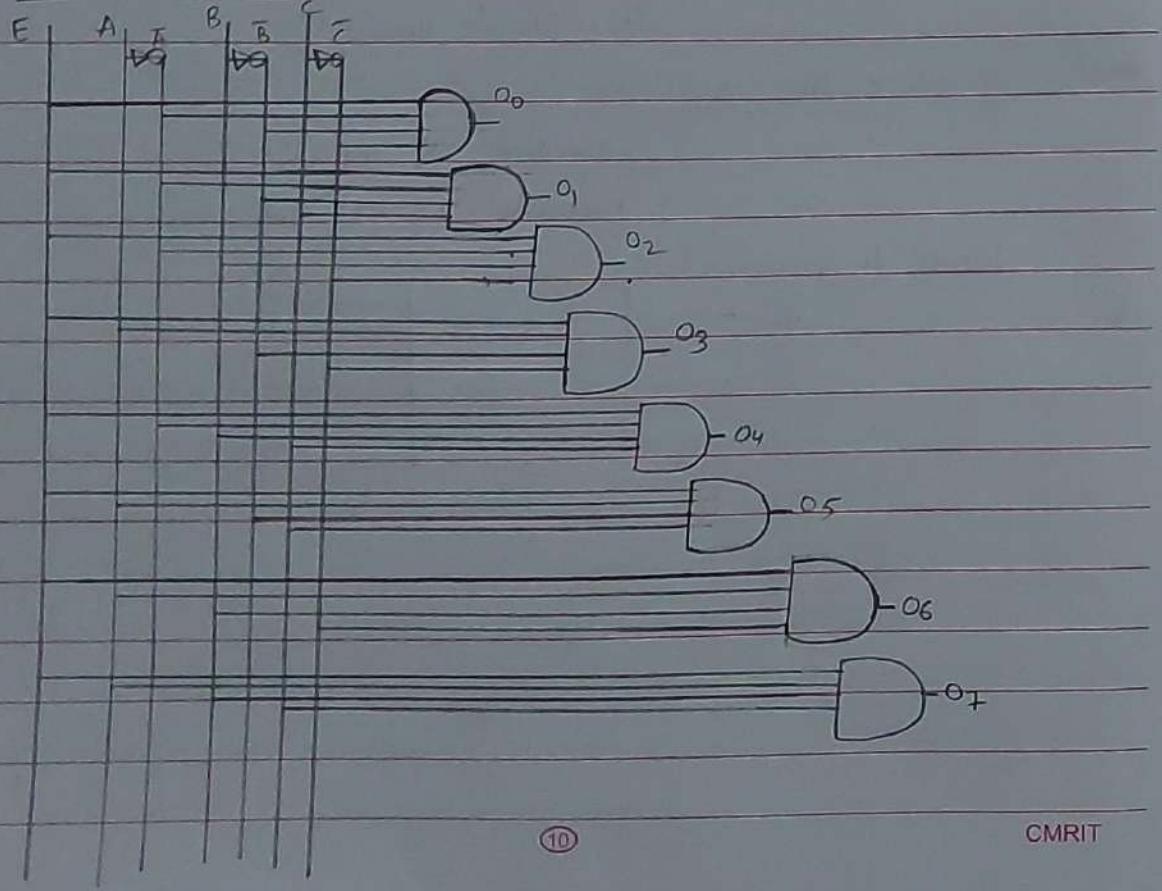


Truth table:-

E	A	B	C	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

output is $O_0 = E\bar{A}\bar{B}\bar{C}$; $O_1 = E\bar{A}\bar{B}C$, $O_2 = E\bar{A}B\bar{C}$, $O_3 = EAB\bar{C}$,
$O_4 = \bar{A}BC$, $O_5 = EA\bar{B}C$, $O_6 = EAB\bar{C}$, $O_7 = EABC$

Circuit diagram:-



4. How does a JK flipflop differs from a S-R flipflop in its basic operations? Explain.

Both J-K (Jack Kilby) flipflops and S-R (set-Reset) flip flops are types of bistable multivibrators used in DLD & CO when they share the ability to store a state, they differ in their basic operations and characteristics

S-R flipflop:-

1) Basic Equation:- The S-R flipflop has two inputs set (S) and Reset (R)

⇒ It is designed to eliminate the ambiguous that can occur in the R-S flipflop when both inputs are active(1).

⇒ The S input sets the flipflop to the state 1, while the R input resets to the state 0.

2) Ambiguous State Handling:- In a S-R flipflop, if both S and R inputs are active simultaneously (1), the behaviour is undefined and it can head to the "meta stable" state.

3) Clock input:- Some implementation of S-R flip-flops include a clock input, allowing the flip-flop to change its state only on a clock Edge.

J-K flip flop:-

⇒ The J-K flipflop also has two inputs: J and K.

⇒ The eliminates the ambiguous state of the S-R flip flop by introducing a "toggle" functionality.

→ when J=1 & K=0, the flip-flop is set.
When J=0 & K=1 the flip-flop is reset.
When both J and K are 1, the flip-flop taggles or changes its state.

2) Ambiguous state Handling:-
→ The J-K flipflop inherently handles the ambiguous state issue by planning a defined behaviour when both J & K inputs are active.

3) Clock input:-
Similar to the S-R flipflop, J-K flip-flops often includes a clock input for synchronous operation.

5. b. Explain about Basic Computer Register?
A Basic computer register is a small unit of memory within the CPU that is used for Temporary storage of data during processing. Registers are fundamental components in CPU operations, such as arithmetic, logic and data movement.

1) Size and Numbers:- Registers are small fast locations typically capable of holding a fixed no. of bits.
→ A CPU contains multiple registers with different purposes such as data registers, address registers and control registers.

2) Data storage:- Registers store binary data in the forms of bits.

→ Data Registers are used for holding Operands, intermediate results.

3) Operations:- Registers are involved in various CPU operations, including Arithmetic & logic operations.

→ They serve as the source & destination for data during these operations.

4) Registers in Instruction Executions:- During the Execution of machine instructions, data is often transfered b/w registers & other parts of the CPU, such as the arithemetic logic unit.

5) Address Registers:- Some registers store memory address & facilitate memory access during program Execution.

6) control Registers:- control register are used for managing the operation of the CPU, controlling interupts and strong status information.

7) Fast Access:- Registers are located within the CPU itself, providing fast access of data compared to main memory.

8) Temporary storage:-

Registers are temporary store locations used for short-term data storage during the program Execution.