

Boolean Algebra

➤ Boolean Algebra rules

$$1) x + 0 = x$$

$$4) x \cdot 1 = x$$

$$7) x + \bar{x} = 1$$

$$2) x + 1 = 1$$

$$5) x + x = x$$

$$8) x \cdot \bar{x} = 0$$

$$3) x \cdot 0 = 0$$

$$6) x \cdot x = x$$

$$9) (\bar{\bar{x}}) = x$$

$$10) \text{ Commutative law } \rightarrow A + B = B + A$$

$$A \cdot B = B \cdot A$$

$$11) \text{ Associative law } \rightarrow (A+B)+C = A+(B+C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

$$2) A(A+B) = A$$

$$A \cdot A + A \cdot B = A$$

$$* 12) \text{ Distributive law } \rightarrow A \cdot (B+C) = A \cdot B + A \cdot C$$

$$A + B \cdot C = (A+B)(A+C)$$

$$A + A \cdot B = A$$

$$* 13) \text{ Redundance law } \rightarrow 1) A + A \cdot B = A$$

$$1) A + AB = A$$

$$2) A \cdot (A+B) = A$$

$$A \cdot 1 + A \cdot B = A$$

$$A(1+B) = A$$

$$14) x + \bar{x}y = x + y$$

$$A \cdot 1 = A$$

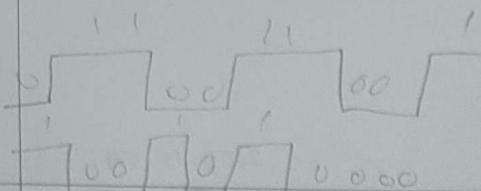
$$A = A$$

* Algebra associated with binary numbers is called Boolean Algebra.

* Variables used in Boolean algebra are called Boolean Variables.

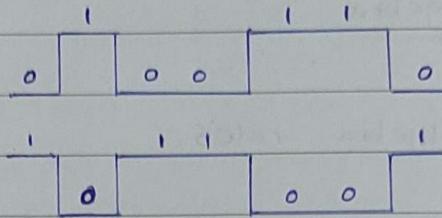
➤ Boolean variable.

* There only can have either 1 or 0 only. "AND, OR, NOT".



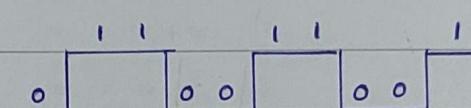
• NOT operation

x	$\text{NOT}(x) / \bar{x}$
0	1
1	0



• AND operation

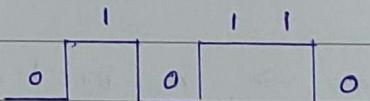
x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1



And

• OR operation

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1



OR

(b)	A	B	$A \cdot B$	$A + A \cdot B$
	0	0	0	0
	0	1	0	0
	1	0	0	1
	1	1	1	1

• perfect Induction.

$$1) x + xy = x$$

$$2) x(x+y) = x$$

x	y	xy	x+xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

x	y	(x+y)	x(x+y)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

Right hand = Left hand.

Left had side = Right hand side.

$$\left\{ \begin{array}{l} \text{2 no. of variables} \\ = 2^2 = 4 \end{array} \right\}$$

$$3) x + \bar{x}y = x + y$$

x	y	\bar{x}	$\bar{x}y$	$x + \bar{x}y$	$x + y$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Left hand side = Right hand side.

Q) prove the above using boolean rules.

$$\begin{aligned} 1) x + xy &= x \\ x(\bar{x} + y) &= x \\ x \cdot 1 &= x \\ x &= x \end{aligned}$$

$$\begin{aligned} 2) x(x+y) &= x \\ x \cdot x + x \cdot y &= x \\ x + x \cdot y &= x \\ x(\bar{x} + y) &= x \\ x \cdot 1 &= x \\ x &= x \end{aligned}$$

$$1) x + \bar{x}y = x + y$$

$$x + \bar{x}y \quad ((x = x + xy))$$

$$x + xy + \bar{x}y$$

$$x + y(x + \bar{x})$$

$$x + y \cdot 1$$

$$x + y = x + y$$

* Using Boolean algebra, Simplify the expressions.

$$1) A = \bar{x}y\bar{z} + \bar{x}yz + x\bar{y}z + xy\bar{z}$$

$$= \bar{x}y(\bar{z} + z) + xz(\bar{y} + y)$$

$$= \bar{x}y(1) + xz(1)$$

$$A = \bar{x}y + xz$$

$$2) x = AB + A(B+C) + B(B+C)$$

$$= AB + AB + AC + BB + BC$$

$$= AB + AC + B + BC$$

$$= AC(B+C) + B(A+1+C)$$

$$= AB + ACB + BC$$

$$= AB + BC$$

$$3) A = x + \bar{y} + \bar{x}y + (x + \bar{y}) \cdot \bar{x}y$$

$$= x + \bar{y} + \bar{x}y + \bar{x}y \cdot x + \bar{x}y \cdot \bar{y} \rightarrow x + \bar{y} + \bar{x}y + 0 \cdot \bar{x} + 0 \cdot y$$

$$= x + \bar{y} + y$$

$$= 1$$

Do this using distributive law,

$$x + \bar{y} + \bar{x}y = x + \bar{y} + \bar{x}y \rightarrow (\bar{y} + \bar{x})(\bar{y} + y)$$

$$x + (\bar{y} + \bar{x}) + (\bar{y} + y)$$

12.10.2022.

4) $Z = ABC [AB + \bar{C} (BC + AC)]$

- = $ABC [AB + BC\bar{C} + AC\bar{C}]$
- = $ABC [AB + 0 + 0]$
- = $ABCAB$
- = ABC

5) $Y = (A + \bar{A}) (AB + A\bar{B})$

- = 1. ($AB + A\bar{B}$)
- = $AB (1 + \bar{C})$
- = $AB (1)$
- = AB .

6) $Y = (A + \bar{B}), (A + C)$

- = $AA + AC + \bar{B}A + \bar{B}C$
- = $A (1 + C) + \bar{B} (A + C)$
- = $A + \bar{B}A + \bar{B}C$
- = $A (1 + \bar{B}) + \bar{B}C$
- = $A + \bar{B}C$

7) $Z = AB + ABC + ABCD + ABCDE$

- = $AB (1 + C + CD + CDE)$
- = $AB ((1 + C) + CD (1 + E))$
- = $AB (1 + CD)$
- = AB

Prove the following

1) $\bar{x}y + \bar{y}z + yz = \bar{x}y + z$ 2) $xz + \bar{x}yz = xz + yz$

$$\begin{aligned} & \bar{x}y + z(\bar{y} + y) && z(x + \bar{x}y) \\ & \bar{x}y + z && z(x + y) \\ & & & xz + zy = xz + yz \\ & & & R.H.S. = L.H.S. \end{aligned}$$

$$\begin{aligned}
 & \text{1) } A\bar{C}\bar{D} + A\bar{B}\bar{D} + A\bar{C}\bar{D} = L.H.S \\
 & \quad A(\bar{C}\bar{D} + \bar{B}\bar{D} + \bar{C}\bar{D}) \\
 & \quad A(\bar{D}(\bar{C} + \bar{C}) + \bar{B}\bar{D}) \\
 & \quad A(\bar{D} + \bar{B}\bar{D}) \\
 & \quad A(\bar{B} + \bar{D}) \\
 & \quad \bar{A} = L.H.S. \\
 & R.H.S = L.H.S.
 \end{aligned}$$

\rightarrow De Morgan's Law. $\rightarrow \cdot \overline{x+y} = \overline{x} \cdot \overline{y}$ $\cdot \overline{xy} = \overline{x} + \overline{y}$

$$\begin{aligned}
 1) \quad & \overline{A + \bar{B}} & 2) \quad & \overline{AB + \bar{CD}} & 3) \quad & (\overline{AB})(\overline{CD}) \\
 & \bar{A} \cdot \bar{\bar{B}} & & (\overline{AB}) \cdot (\overline{CD}) & & (\overline{AB}) + (\overline{CD}) \\
 & \bar{A}\bar{B} & & (\bar{A} + \bar{B}) \cdot (\bar{C} + \bar{D}) & & AB + CD \\
 & & & \bar{A}\bar{C} + \bar{A}\bar{D} + \bar{B}\bar{C} + \bar{B}\bar{D} & &
 \end{aligned}$$

\rightarrow Apply De Morgan's Theorems to simplify the given expression.

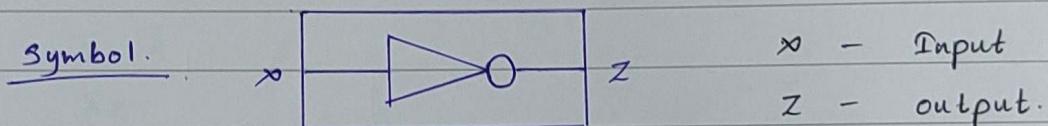
$$\begin{aligned}
 1) \quad & \overline{(A + \bar{B})(\bar{C} + \bar{D})} & 3) \quad & \overline{A + \bar{B}\bar{C} + \bar{C}\bar{D} + \bar{B}\bar{C}} \\
 & (\overline{A + \bar{B}}) + (\overline{\bar{C} + \bar{D}}) & & \bar{A} \cdot \overline{\bar{B}\bar{C}} \cdot \overline{\bar{C}\bar{D}} + \bar{B}\bar{C} \\
 & \bar{A} + \bar{\bar{B}} + \bar{\bar{C}} + \bar{\bar{D}} & & \bar{A} \cdot \bar{B}\bar{C} \cdot (\bar{C} + \bar{D}) + \bar{B}\bar{C} \\
 & \bar{A}\bar{B} + \bar{C}\bar{D} // & & \bar{A}\bar{B}\bar{C}(\bar{C} + \bar{D}) + \bar{B}\bar{C} \\
 & & & \bar{A}\bar{B}\bar{C}\bar{C} + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{B}\bar{C} \\
 2) \quad & \overline{\bar{x}\bar{y}(\bar{w} + \bar{y})} & & \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{B}\bar{C} \\
 & \bar{x}\bar{y} + \bar{w} + \bar{y} & & \bar{A}\bar{C}(\bar{B} + \bar{B}\bar{C}) + \bar{B}\bar{C} \\
 & \bar{x} + \bar{w} // & & \bar{A}\bar{C}B + \bar{B}\bar{C} \\
 & & & B(\bar{A}\bar{C} + C) \\
 & \overline{\bar{A} \cdot \bar{B}} + \overline{B \cdot \bar{A}} & & B(\bar{A} + C) \\
 & A \cdot \bar{A}\bar{B} + B \cdot \bar{A}\bar{B} & & \bar{B}\bar{A} + BC // \\
 & A(\bar{A} + \bar{B}) \cdot B(\bar{A} + \bar{B}) & &
 \end{aligned}$$

Basic logic gates.

- * Hardware components and it's of intergrated circuit.
- * It is a digital circuit that performs common logic / Boolean functions such as AND / OR / NOT.

1) NOT gate (Inverter)

- * performs logical complementation.



$$\text{operation: } z = \overline{x}$$

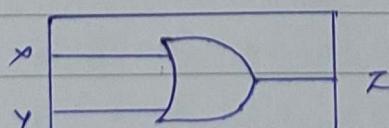
Truth table.

x	z
0	1
1	0

2) OR gate

- * performs logical addition.

Symbol.



x, y - Input

z - output

$$\text{operation: } z = x + y$$

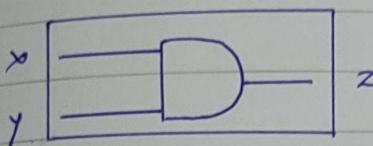
truth table.

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

b) AND gate.

- performs logical multiplication.

Symbol



x, y - input

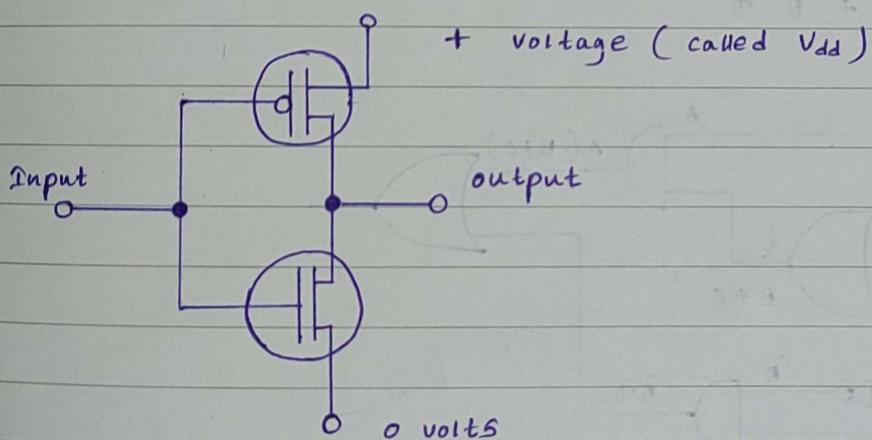
z - output

truthtable

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

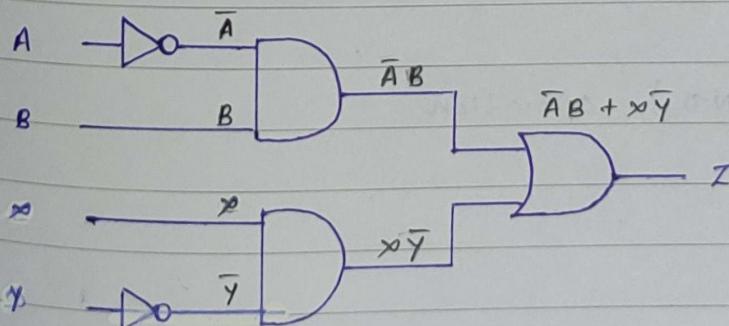
operation: $z = x \cdot y$

Implement not gate using transistors?



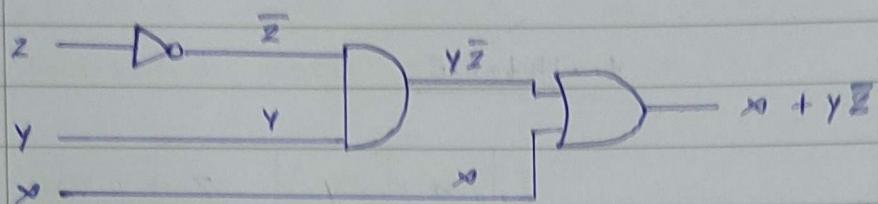
Implement the circuit for the following expression using AND, OR, NOT gates.

$$\{ z = \bar{A}B + x\bar{y} \}$$

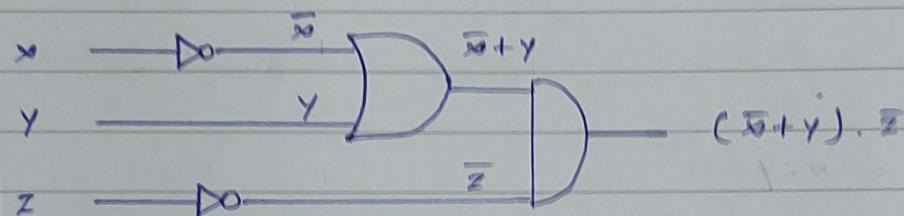


➤ Implement following boolean expressions using logic gates.

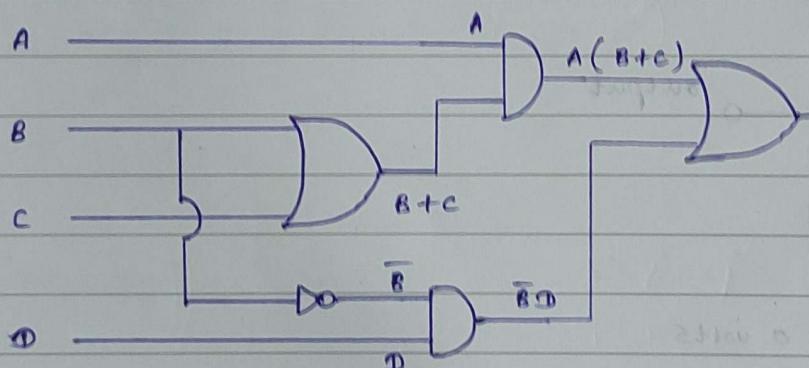
1) $A = x + y\bar{z}$



2) $A = (\bar{x} + y)\bar{z}$



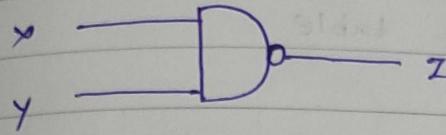
3) $Z = A(B+C) + \bar{B} \cdot \bar{D}$



➤ other logic gates.

1) NAND gate

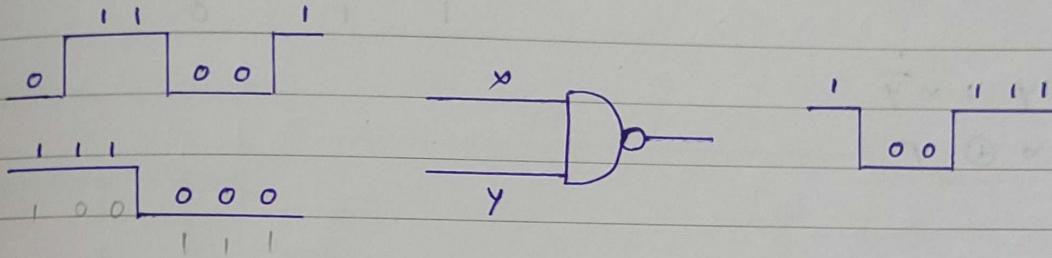
* performs NOT (AND) operation.

Symbol.

Truth table.

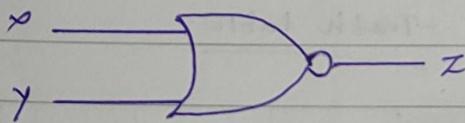
x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

operation: $\overline{x \cdot y}$
 $= \text{not}(x \cdot y)$

ex:-


2) NOR gate.

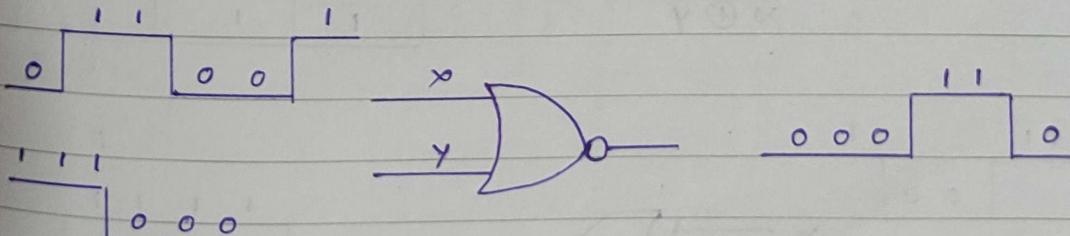
* performs NOT (OR) operation.

Symbol.

Truth table.

x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

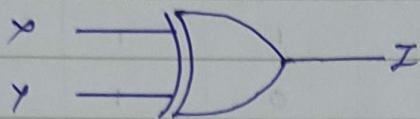
operation: $\overline{x+y}$
 $= \text{not}(x+y)$

ex:-



3) Exclusive OR gate (XOR)

Symbol



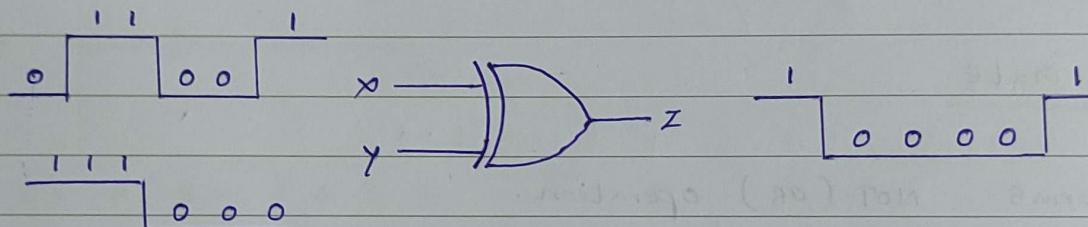
Truth table

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

operation:

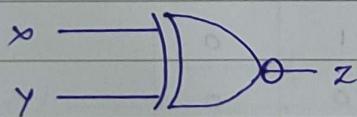
$$\begin{aligned} z &= \overline{x}y + x\overline{y} \\ &= x \oplus y \end{aligned}$$

ex:-



4) Exclusive NOR gate (XNOR)

Symbol

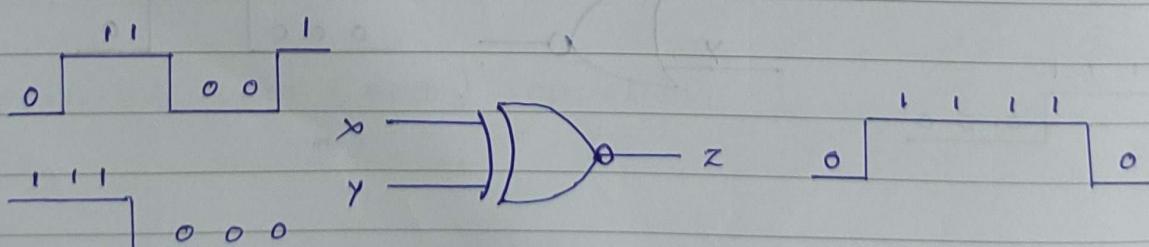


Truth table

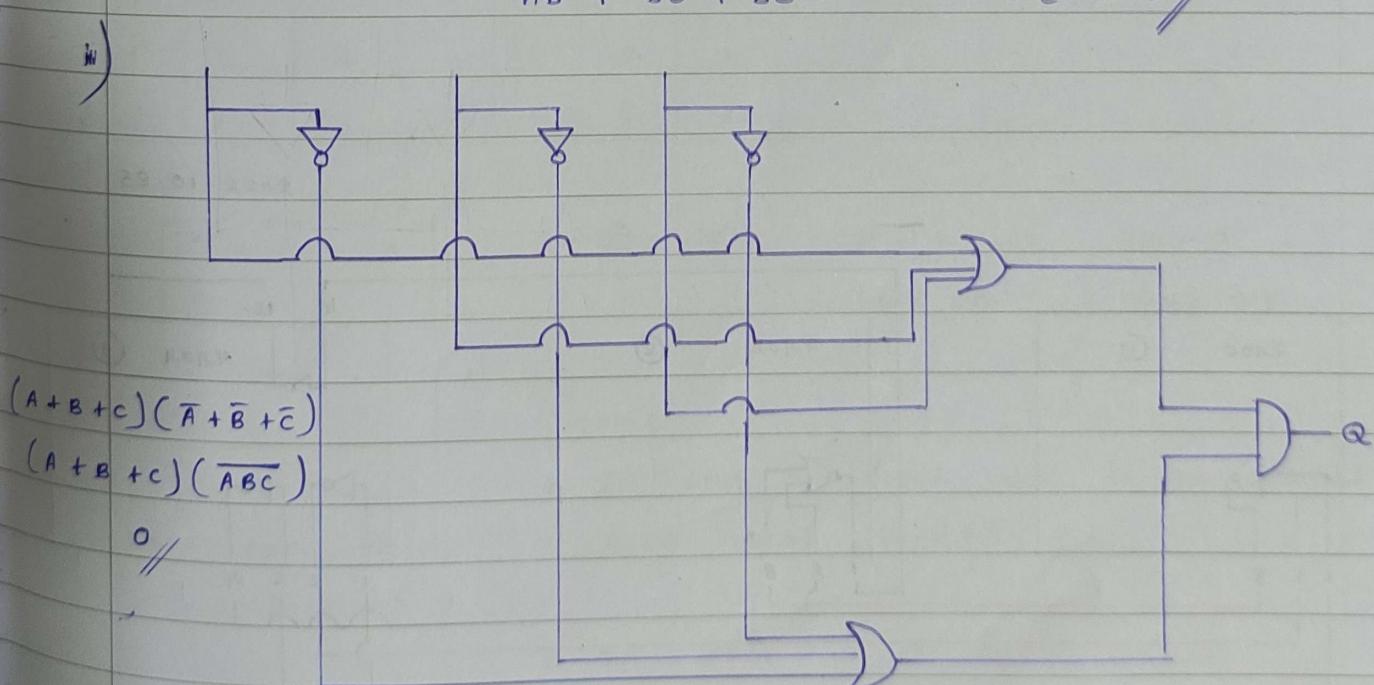
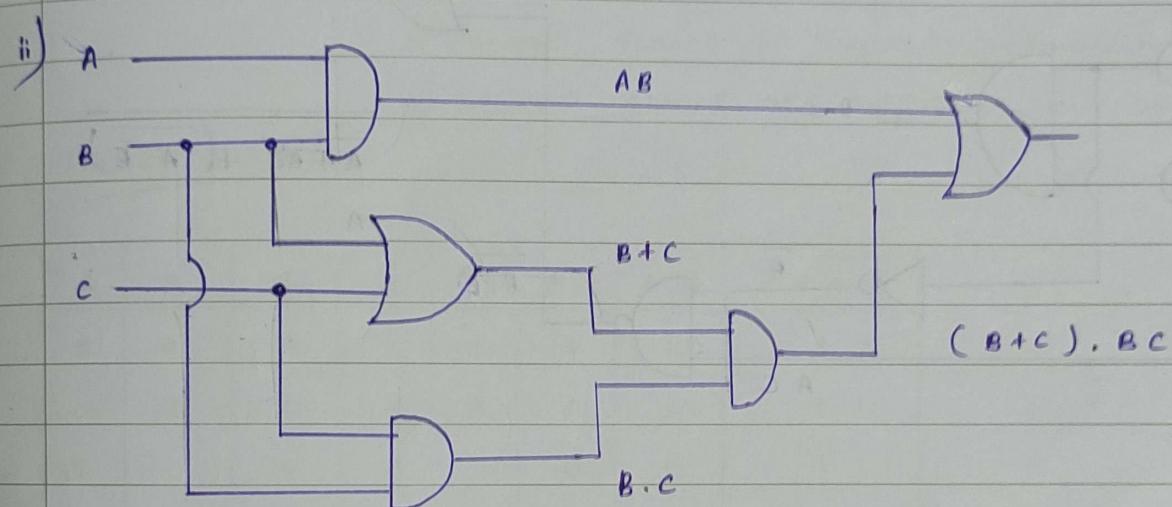
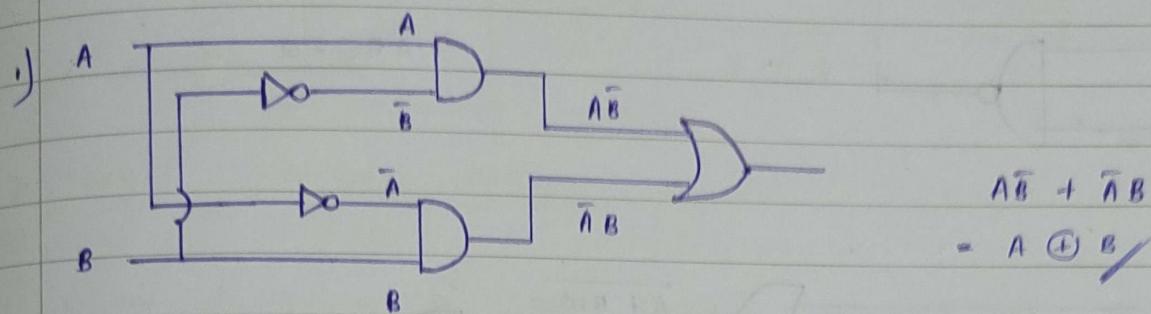
\overline{x}	\overline{y}	\overline{xy}	x	y	z	xy
1	1	1	0	0	1	0
1	0	0	0	1	0	1
0	1	0	1	0	0	0
0	0	1	1	1	1	1

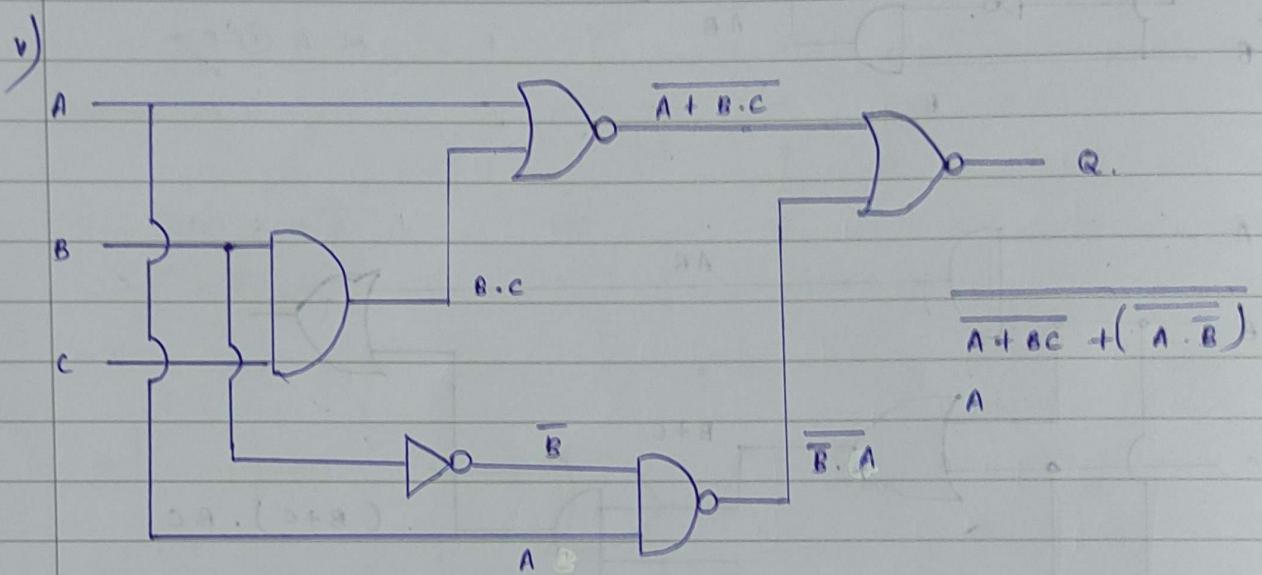
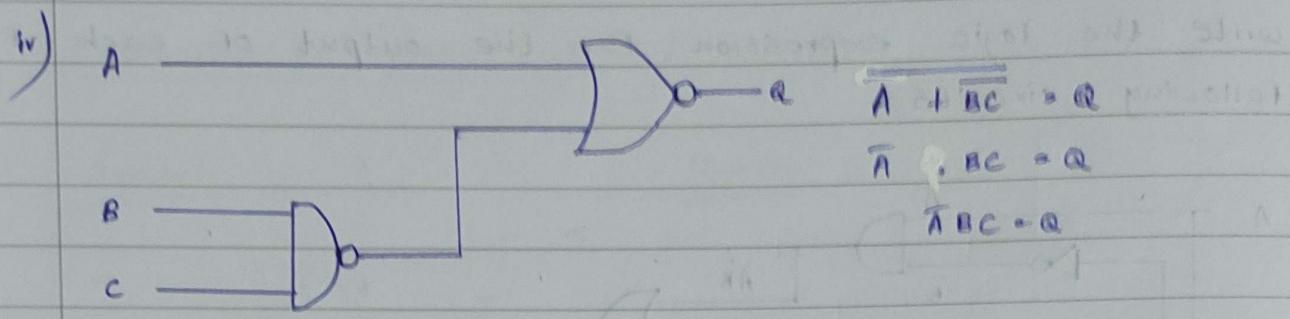
$$\begin{aligned} \text{operation: } z &= \overline{x}\overline{y} + x\overline{y} \\ &= \overline{x \oplus y} \end{aligned}$$

ex:-

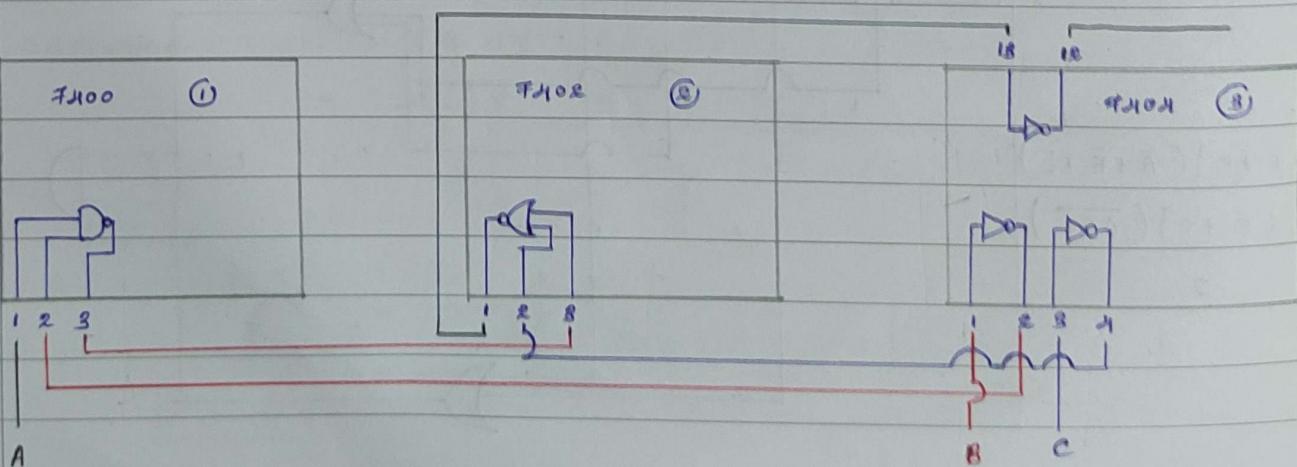


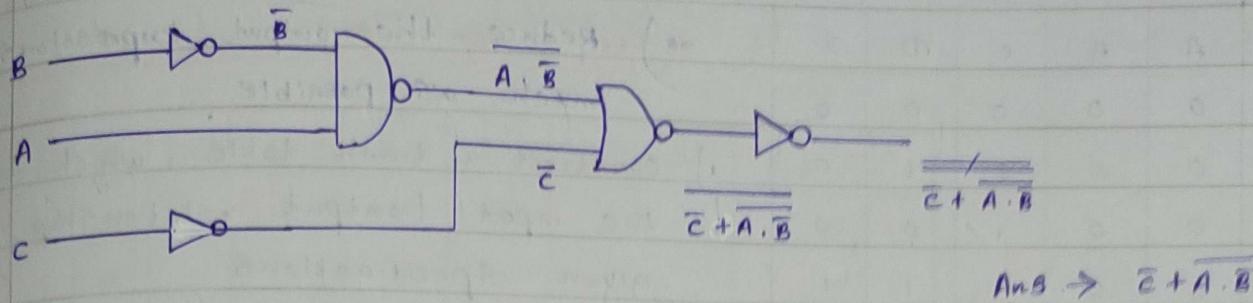
* write the logic expression for the output of each of the following circuits.





Page. 10. 25





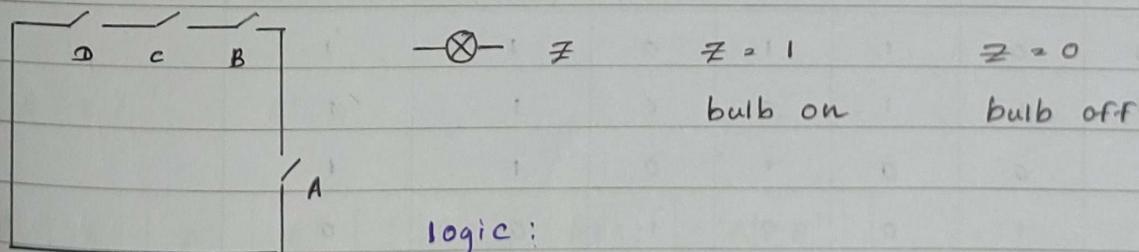
> Logic circuit design.

when designing a logic circuit, we work with 2 sets of known values,

- 1) Various states of 'input' combination of a circuit.
- 2) Various states of 'output' " of each input.

Truth table:

A	B	C	\bar{B}	\bar{C}	$A \cdot \bar{B}$	$\bar{A} \cdot \bar{B}$	$\bar{C} + \bar{A} \cdot \bar{B}$
0	0	0	1	1	0	1	1
0	0	1	1	0	0	1	1
0	1	0	0	1	0	1	1
0	1	1	0	0	0	1	1
1	0	0	1	1	1	0	1
1	0	1	1	0	1	0	0
1	1	0	0	1	0	1	1
1	1	1	0	0	0	1	1



logic:

* If any 2 doors are to be opened
Door open $\rightarrow 1$ Bulb must turned ON.
close $\rightarrow 0$

* If door A is open without
considering other doors bulb should
Turn ON.

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	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

• Steps for design a logic circuit

much as possible.

1) construct a truth table , which describes the input / output relationship by the given specifications.

2) Derive an algebraic expression for each of the output.

- a) SOP method
- b) POS method.

3) Reduce the output expression as much as possible.

- a) algebraically
- b) using K-maps
- c) using Tabular method.

4) Implement the circuit using gates.

Tutorial 1

2022.10.26.

Q1) i) $(A \cdot B)' = A' + B'$

A	B	A'	B'	$A \cdot B$	$(A \cdot B)'$	$A' + B'$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

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

A	B	C	$B+C$	$A \cdot B$	$A \cdot C$	$A \cdot (B+C)$	$A \cdot B + A \cdot C$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

4) $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

A	B	C	$A \cdot B$	$B \cdot C$	$(A \cdot B) \cdot C$	$A \cdot (B \cdot C)$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

4) $(A+B)' = A' \cdot B'$

A	B	A'	B'	$(A+B)$	$(A+B)'$	$A' \cdot B'$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

$$6) (A+B)+C = A+(B+C)$$

A	B	C	$(A+B)$	$(B+C)$	$(A+B)+C$	$A+(B+C)$
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	0	1	1
0	1	1	1	1	1	1
1	0	0	1	0	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

02) Apply De Morgan's law.

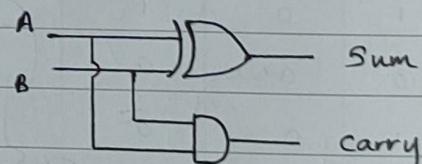
$$\overline{xy} = \overline{x} + \overline{y}$$

$$\begin{aligned}\overline{A \cdot BC} &= \overline{\overline{A}} + \overline{BC} \\ &= \overline{A} + \overline{B} + \overline{C}\end{aligned}$$

2022.09.01

Binary half adder

Truth table.



A	B	carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

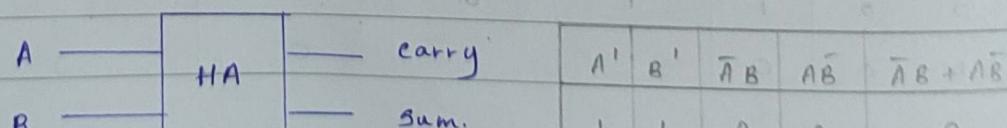
$$\text{Sum} = \overline{AB} + \overline{A}\overline{B} = A \oplus B \rightarrow \text{xor}$$

$$\text{carry} = AB \rightarrow \text{AND}$$

A, B \rightarrow Inputs

carry, sum \rightarrow outputs.

* The combination of output's gives the binary half adder.



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

$\boxed{10} \leftarrow @$ $\boxed{00} \leftarrow 0$
 $\boxed{11} \leftarrow 3$ $\boxed{01} \leftarrow 1$

$0+0 \rightarrow 0 \rightarrow 00$

$0+1 \rightarrow 1 \rightarrow 01$

$1+0 \rightarrow 1 \rightarrow 01$

$1+1 \rightarrow 2 \rightarrow 10$

halt

Adders

Binary full adder.

* It is a logic circuit that can add three inputs, 1-bit binary numbers.

			2^1	2^0						
	A	B	C	carry	sum	AB	BC	AC		
$\bar{A}\bar{B}C \rightarrow$	0	0	0	0	0	0	0	0	$0+0+0 \rightarrow 0$	00
$\bar{A}\bar{B}\bar{C} \rightarrow$	0	0	1	0	1	0	0	0	$0+0+1 \rightarrow 1$	01
$\bar{A}B\bar{C} \rightarrow$	0	1	0	0	1	0	0	0	$0+1+0 \rightarrow 1$	01
$A\bar{B}\bar{C} \rightarrow$	0	1	1	1	$\bar{A}\bar{B}C$	0	0	1	$0+1+1 \rightarrow 2$	10
$A\bar{B}\bar{C} \rightarrow$	1	0	0	0	1	0	0	0		1
	1	0	1	1	$\bar{A}\bar{B}C$	0	0	0		2
	1	1	0	1	$\bar{A}B\bar{C}$	0	1	1		2
$ABC \rightarrow$	1	1	1	1	$A\bar{B}C$	1	1	1		$3 \rightarrow 11$

$$\text{Sum} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$= \bar{A}(\bar{B}C + B\bar{C}) + A(B\bar{C} + BC)$$

$$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$$

$$= \bar{A} \otimes + A \otimes$$

$$= A \oplus \otimes$$

$$= A \oplus B \oplus C$$

$$\text{carry} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$= BC(\bar{A} + A) + A\bar{B}C + AB\bar{C}$$

$$= BC + A\bar{B}C + AB\bar{C}$$

$$= C(B + A\bar{B}) + AB\bar{C}$$

$$= C((A+B)(B+\bar{B})) + AB\bar{C}$$

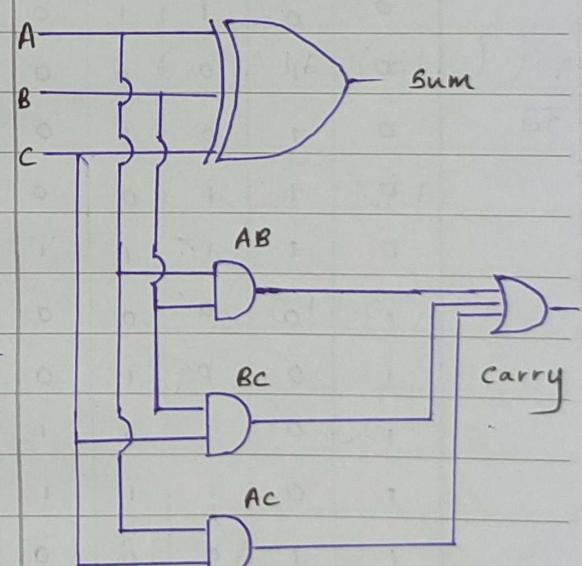
$$= AC + BC + A\bar{B}C$$

$$= A(C + B\bar{C}) + BC$$

$$= A((B+C)(C+\bar{C})) + BC$$

$$= AB + AC + BC$$

$$= AB + BC + AC$$



- Q) 1) Design a logic circuit to add two, 2-bit binary numbers.
 Implement the circuit using logic gates.
- 2) In a digital system, an error comparator is required which will compare two, 2-bit binary numbers A, B and gives separate outputs for the conditions $A = B$, $A > B$ and $A < B$. Design a logic circuit to perform the above functions.

				x	y	z	
				A_1	A_0	B_1	B_0
1)	A_1			2^2	x	1 1	1 1
	A_0			2^1	y	3 3	$3 + 3 = 6$
	B_1			2^0	z	6	$2^2x_1 + 2^1x_1 + 2^0x_0$
	B_0						$4 + 2 + 0 = 6$
3	$\leftarrow (2^1 \quad 2^0) \quad (2^1 \quad 2^0) \rightarrow 3$						$=$

A_1	A_0	B_1	B_0	x	y	z	Input \rightarrow
0	0	0	0	0	0	0	
0	0	0	1	0	0	1	$A_1 + A_0 + B_1 + B_0$
0	0	1	0	0	1	0	$= 2^1 + 2^0 + 2^1 + 2^0$
0	0	1	1	0	1	1	$= 2 + 1 + 2 + 1$
0	1	0	0	0	0	1	$= 6$
0	1	0	1	0	1	0	
0	1	1	0	0	1	1	output \rightarrow
0	1	1	1	1	0	0	
1	0	0	0	0	1	0	
1	0	0	1	0	1	1	
1	0	1	0	1	0	0	
1	0	1	1	1	0	1	
1	1	0	0	1	1	1	
1	1	0	1	1	0	0	
1	1	1	0	1	0	1	
1	1	1	1	1	1	0	

$$X = A_1 B_1 + A_1 A_0 B_0 + A_0 B_1 B_0$$

2022. 11. 08

0

1

10

$$Y = \overline{A_1} \overline{A_0} B_1 + \overline{A_1} \overline{B_1} \overline{B_0} + A_1 \overline{B_1} \overline{B_0} + A_1 \overline{A_0} \overline{B_1} + \overline{A_1} A_0 \overline{B_1} B_0 + A_1 A_0 B_1 B_0$$

$$Z = A_0 \oplus B_0$$

100

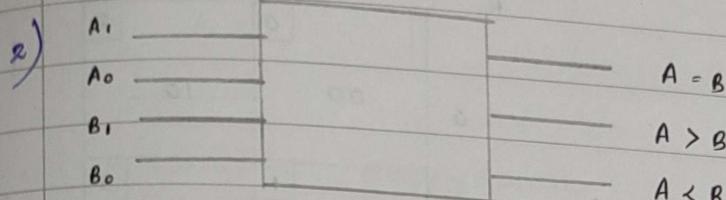
101

110

111

1000

1001



A ₁	A ₀	B ₁	B ₀	A = B	A > B	A < B	
0	0	0	0	1	0	0	101
0	0	< 0	1	0	0	1	1100
0	0	< 1	0	0	0	1	1101
0	0	< 1	1	0	0	1	1110
0	1	> 0	0	0	1	0	1111
0	1	> 0	1	1	0	0	
0	1	= 1	0	0	0	1	
0	1	= 1	1	0	0	1	
1	0	0	0	0	1	0	
1	0	0	1	1	1	0	
1	0	< 1	0	1	0	0	
1	0	< 1	1	0	0	1	
1	1	> 0	0	0	1	0	
1	1	> 0	1	0	1	0	
1	1	= 1	0	0	1	0	
1	1	= 1	1	1	0	0	

> Karnaugh maps (K-maps)

- A Boolean expression. (2 variables)

		0	1
		$\bar{A}\bar{B}$	$\bar{A}B$
0	0	$\bar{A}\bar{B}$	$\bar{A}B$
	1	$\bar{A}B$	AB

cell

		0	1
		00	10
0	0	0	2
	1	1	3

cell number.

example :-

$$x = AB + \bar{A}B + \bar{B}A$$

1 → A or B

0 → \bar{A} or \bar{B}

		0	1
		0	1
0	0	0	1
	1	1	1

1) How to find the value

2) How to find the cell number

$$\begin{array}{l} A \\ \bar{A} \end{array} \begin{array}{l} B \\ \bar{B} \end{array} \rightarrow 00 \rightarrow 0$$

$$\begin{array}{l} A \\ \bar{A} \end{array} \begin{array}{l} B \\ \bar{B} \end{array} \rightarrow 01 \rightarrow 1$$

$$1+0 \rightarrow 10 \rightarrow 2$$

$$1+1 \rightarrow 11 \rightarrow 3$$

only one variable change at a time

- B variable expression.

		00	01	10	11
		$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$A\bar{B}\bar{C}$	$A\bar{B}C$
0	0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$A\bar{B}\bar{C}$	$A\bar{B}C$
	1	$\bar{A}B\bar{C}$	$\bar{A}BC$	$A\bar{B}\bar{C}$	ABC

* 11 comes before 10

$$x = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$$

ex:-

	AB	00	01	11	10
C	0	0	0	0	1
	1	1	0	1	0

4 variable expression.

	AB	00	01	11	10
CD	00	$A\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$
	01	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$
	11	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$
	10	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$

ex:-

	AB	00	01	11	10	
CD	00	0	4	12	8	
	01	1	5	13	9	
	11	3	7	15	11	
	10	0	1	0	0	
		2	6	14	10	
		0	0	0	0	

$$x = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$$

- Draw K-maps for the following SOP expressions.

i) $Z = \bar{A}B\bar{C} + A\bar{B}C + ABC$

		AB	00	01	11	10
		C	0	0	1	1
			0	1	0	0
x						

$A \rightarrow 0 / C \rightarrow 1$

on 00 11

$A \rightarrow 1 / B \rightarrow 0$

on 01 10

min variable on 01

max value on 11

max 11

ii) $Z = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}CD$

		AB	00	01	11	10
		C	00	0	0	0
			01	0	1	0
x						

i) $Z = \bar{A}C + A\bar{B}$

$\bar{A}C = \bar{A}C \cdot 1$

$= \bar{A}C \cdot (B + \bar{B})$

$= \bar{A}CB + \bar{A}C\bar{B}$

$= \bar{A}BC + \bar{A}\bar{B}C$

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

		AB	00	01	11	10
		C	0	0	0	1
			1	1	1	0
x						

$A\bar{B}\bar{C}$ $\bar{A}BC$

$\bar{A}\bar{B}\bar{C}$ ABC

$A\bar{C}B$

ii) $Z = AB + \bar{A}\bar{B}C$

		AB	00	01	11	10
		C	0	0	1	1
			1	1	1	1
						0
						0

iii) $Z = \bar{A}\bar{C}\bar{D} + BC$

		AB	00	01	11	10
		CD	00	0	0	0
			01	0	0	0
			11	0	1	1
			10	0	1	1

iv) $Z = A + \bar{B}\bar{C}\bar{D}$

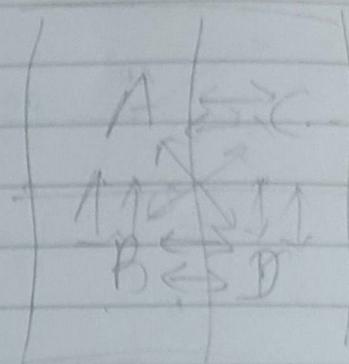
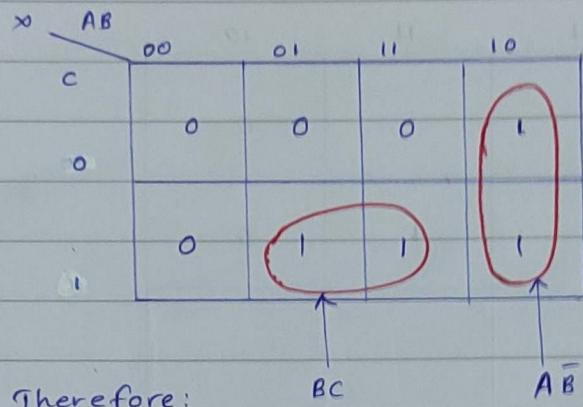
$A\bar{B}\bar{C}\bar{D}$
 $\bar{A}\bar{B}\bar{C}\bar{D}$

		AB	00	01	11	10
		CD	00	1	0	1
			01	0	0	1
			11	0	0	1
			10	0	0	1

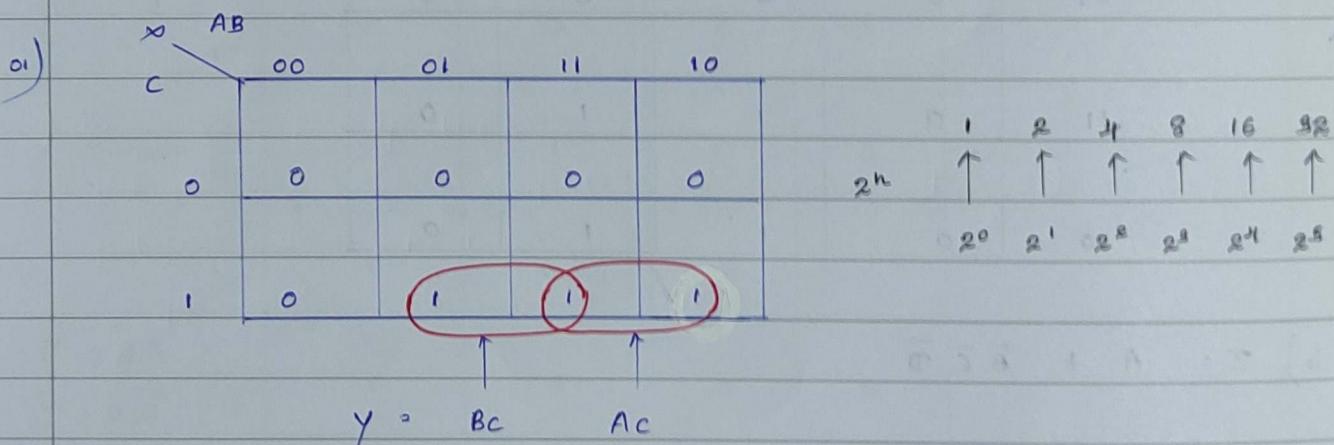
$\bar{A}\bar{B}\bar{C}\bar{D}$
 $\bar{A}\bar{B}\bar{C}\bar{D}$
 $\bar{A}\bar{B}\bar{C}\bar{D}$
 $\bar{A}\bar{B}\bar{C}\bar{D}$
 $A\bar{B}\bar{C}\bar{D}$
 $A\bar{B}\bar{C}\bar{D}$
 $A\bar{B}\bar{C}\bar{D}$
 $A\bar{B}\bar{C}\bar{D}$

> minimal map.

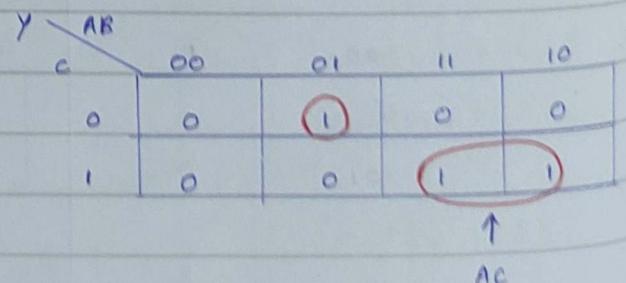
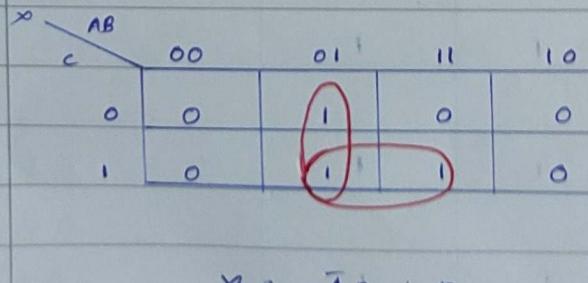
$$\text{ex:- } x = \bar{A}BC + A\bar{B}C + A\bar{B}\bar{C} + ABC$$



$$\bar{A}BC + A\bar{B}C + A\bar{B}\bar{C} + ABC$$



Q) Simplify the following K-maps



	AB	00	01	11	10
C	0	1	0	0	1
	1	0	0	0	1

	AB	00	01	11	10
C	0	1	1	1	1
	1	0	0	0	1

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

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

	AB	00	01	11	10
CD	00	0	0	0	0
	01	0	0	1	1
	11	1	0	0	1
	10	1	0	0	1

	AB	00	01	11	10
CD	00	0	0	0	0
	01	0	0	1	1
	11	1	1	1	1
	10	1	1	1	1

$$Z = A\overline{C}\overline{D} + \overline{B}\overline{C}$$

$$Z = AD + C$$

	AB	00	01	11	10
CD	00	1	1	0	0
	01	1	1	1	0

$$Z = \overline{A} + BC$$

	AB	00	01	11	10
CD	00	1	0	0	1
	01	0	0	0	0
	11	1	1	1	0
	10	1	1	0	1

	AB	00	01	11	10
CD	00	1	1	1	0
	01	1	1	1	0
	11	0	0	1	1
	10	0	0	1	1

$$Z = \overline{B}\overline{D} + \overline{A}C + BCD + \overline{B}C\overline{D}$$

$$\Rightarrow \overline{B}\overline{D} + \overline{A}C + BCD$$

$$Z = AC + \overline{C}B + \overline{A}\overline{C} \quad \text{or}$$

$$AC + AB + \overline{AC}$$

➤ pos expressions.

ex:- $A\bar{B}\bar{C} + \bar{A}BC + ABC \leftarrow \text{sop}$ (some of products)

		AB	C	00	01	11	10
		0	0	0	0	1	
1	1	0		1	1	0	

$Z = A\bar{B}\bar{C} + BC \rightarrow \text{minimal map}$

$$\bar{Z} =$$

$\bar{Y} = \bar{A}\bar{B}B + B\bar{C} + \bar{B}\bar{B}\bar{C} \leftarrow \text{sop expression}$

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

$Y = \bar{A}\bar{B} \cdot \bar{B}\bar{C} \cdot \bar{B}\bar{C}$

$y = (A+B) \cdot (\bar{B}+C) \cdot (B+\bar{C}) \leftarrow \text{pos expression.}$

• Steps

- 1) Draw the K-map for X .
- 2) get the minimal map for \bar{X} (cover all 0's)
- 3) get \bar{X} in sop form.
- 4) complement both sides.

2022. 11. 22.

• get the simplified pos expressions from the following K-maps.

		$x_1 x_2$	x_3	00	01	11	10
		0	1	1	0	0	
1	1	1	1	1	1	0	

$$\bar{Z} = x_1 \bar{x}_3 + x_1 \bar{x}_2$$

$$\bar{\bar{Z}} = \bar{x_1 \bar{x}_3} + \bar{x_1 \bar{x}_2}$$

$$Z = \bar{x}_1 x_3 + \bar{x}_1 x_2$$

$$Z = (\bar{x}_1 + x_3) \cdot (\bar{x}_1 + x_2)$$

pos

expression.

Z

$x_1 x_2$

$x_3 x_4$

	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	1	0	1
10	1	1	1	1

$$\bar{Z} = \bar{x}_2 \bar{x}_3 + \bar{x}_3 x_2 +$$

$$x_1 x_2 x_3 x_4$$

$$\bar{\bar{Z}} = \frac{\bar{x}_2 \bar{x}_3}{x_1 x_2 x_3 x_4} + \frac{\bar{x}_3 x_2}{x_1 x_2 x_3 x_4} +$$

$$Z = (x_2 + x_3) \cdot (x_3 + \bar{x}_2) \\ (\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4)$$

Simplify x, Y, Z given by the following truth table using K-map.

ABC	XYZ
000	100
001	100
010	011
011	011
100	001
101	111
110	110
111	101

$$x = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}C + ABC +$$

$$ABC$$

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + ABC$$

$$Z = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + ABC$$

X

AB

C

	00	01	11	10
0	1	0	1	0
1	1	0	1	1

$$x = \bar{A}\bar{B} + AB + AC$$

Y	AB	00	01	11	10
C	\diagdown	0	1	1	0
		0	1	0	1

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

Z	AB	00	01	11	10
C	\diagdown	0	1	0	1
		0	1	1	1

$$Z = \bar{A}B + A\bar{B} + AC$$

- Find the simplified Sop expressions : *** important example**

Z	AB	00	01	11	10
C	\diagdown	x	0	1	0
		1	1	x	0

$$Z = AB + \bar{A}C$$

$$Z = \bar{Z}m(1, 3, 6, 7)$$

{ Z is a function of A, B, C }

$$I = \bar{Z}m$$

Some of minterms (Sop)

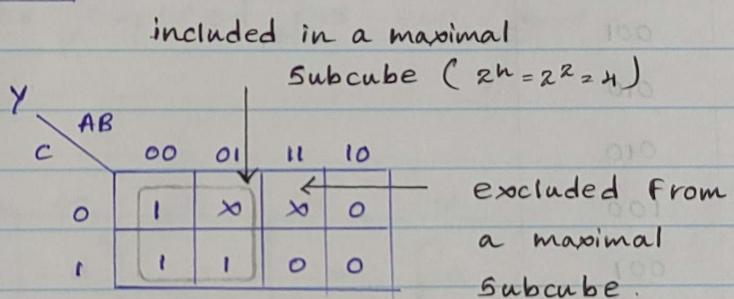
Z	AB	00	01	11	10
CD	\diagdown	0	0	0	1
		0	x	1	1
		0	1	x	0
		x	0	0	0

$$Z = A\bar{B}\bar{C} + B\bar{D}$$

Incompletely Specified boolean functions. ($\text{Don't care} \rightarrow X$)

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	X
1	0	1	X
1	1	0	0
1	1	1	1

* The value of the functions at "Don't care" conditions, is either not significant or those A, B, C values will never occur. Therefore, don't cares can be either Included or excluded in a maximal Subcube so as to get the minimal map.



100 1100

010 0010

001 1010

100 0110

100 1110

0001

010 1001

001 0101

100 1101

010 0011

010 1011

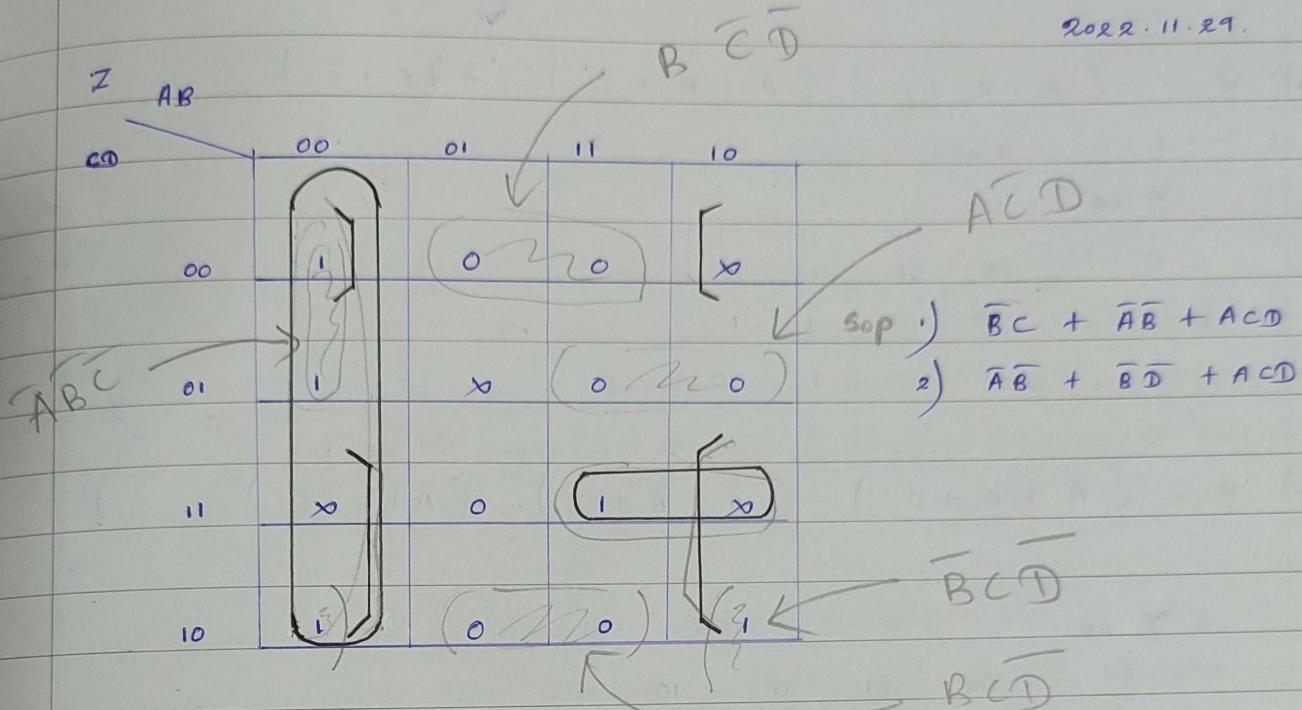
000 0111

001 1111

$$Z = \overline{AB} + \overline{BD} + ACD$$

1 2 4 8 16

2022.11.29.



Get the simplified pos expression.

	Z	AB'	CD
00	0	1	00
01	0	1	01
11	0	x	11
10	1	x	10

$$\begin{aligned} Z &= \overline{BC} + CD \\ Z &= \overline{\overline{BC}} + \overline{CD} \\ &= \overline{\overline{BC}} \cdot \overline{CD} \\ &= (B+C) \cdot (\overline{C} \cdot \overline{D}) \end{aligned}$$

Sop
Minterms and maxterms.

$$Y = \overline{m}(3,4)$$

$$\text{minterm } 3 \rightarrow \overline{ABC}$$

$$\text{minterm } 4 \rightarrow A\overline{B}\overline{C}$$

$$Y_2 = \Gamma IM(3,4)$$

$$\text{maxterm } 3 \rightarrow A + \overline{B} + \overline{C}$$

$$\text{maxterm } 4 \rightarrow \overline{A} + B + C$$

$$Y = (\bar{A}\bar{B}\bar{C}) + (\bar{A}\bar{B}C) + (\bar{A}B\bar{C})$$

pos.) $Y = (A+B+C) \cdot (A+\bar{B}+\bar{C}) \cdot (\bar{A}+B+\bar{C})$

		Y	AB	C		
			00	01	11	10
C	0	0	1	1	1	
	1	1	0	1	0	

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

2) $Y = (A+B+C) \cdot (\bar{A}+\bar{B}+C) \cdot (\bar{A}+B+\bar{C})$

		Y	AB	C		
			00	01	11	10
C	0	0	1	0	1	
	1	1	1	0	1	

a) $Z = f(A, B, C, D)$

$Z = 1$ for the minterms (0, 2, 5, 7, 8, 13)

$Z = 0$ don't care for the minterms (10, 15)

$Z = 0$ for the remaining minterms.

a. Draw the truth table.

b. Simplify Z using K-maps method

c. Draw the circuit Diagram.

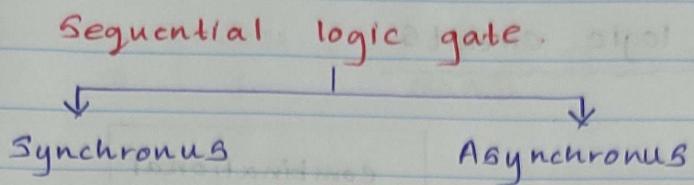
		Z	AB	D		
			00	01	11	10
D	00	1	0	0	1	
	01	1	5	13	9	

		00	01	11	10
		0	1	1	0
D	00	1	0	0	1
	01	1	5	13	9

		00	01	11	10
		0	1	1	0
D	00	1	0	0	1
	01	1	5	13	9

c) $Z = \frac{BD + \bar{B}\bar{D}}{B \oplus D}$





- Synchronous Sequential logic circuit.

* Various block of the logic circuit are triggered by the same clock signal, called the master clock. Therefore, the output of these blocks (devices) change at the same time.

- Asynchronous Sequential logic circuit.

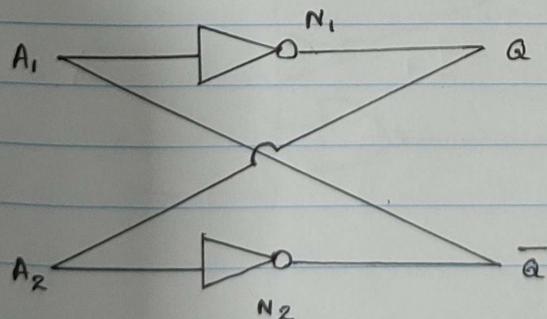
* In this type of circuit, different parts of the circuit are triggered by different signals. Therefore, the output of different block of the circuit can change at different instances.

Latches and flip-flops.

Latch

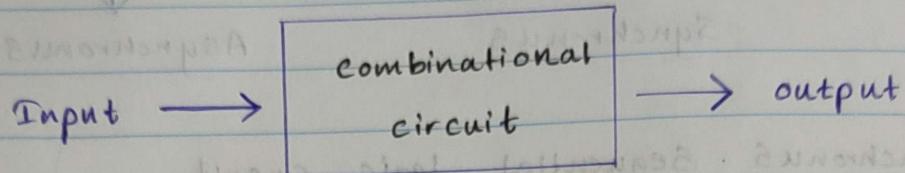
- A latch is a data storage (memory) device that can store one bit of information.

- The basic digital memory circuits is obtained by cross-coupling two NOT gates.



* The output of each gate is connected to the input of the other end and this feedback combination is called latch.

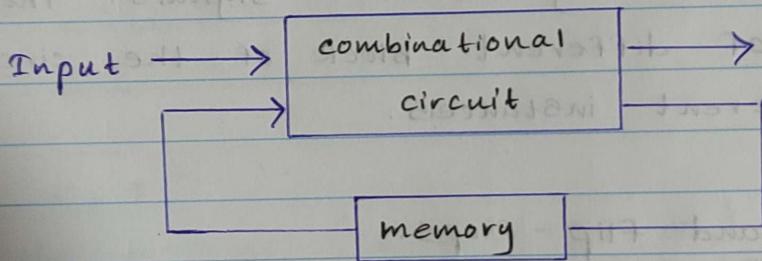
> Sequential logic circuits



A combinational circuit directly consists of several input signals, several output signals and an interconnection of gates. When applying the same inputs, a combinational circuit always produce the same outputs.

- * every input has signal, unique output.

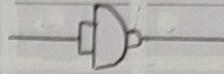
> Sequential circuit



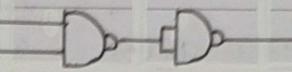
a)	A	B	C	D	Z
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	X
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	X

NAND

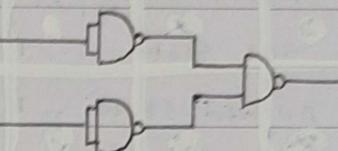
NOT \rightarrow



AND \rightarrow

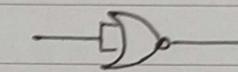


OR \rightarrow

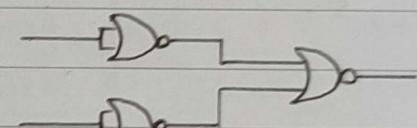


NOR

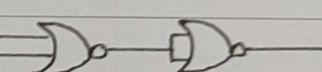
NOT \rightarrow



AND \rightarrow

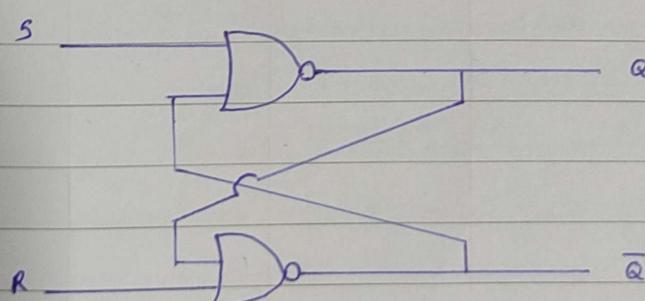


OR \rightarrow



2022.12.06

SR Latch (Set reset latch)



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

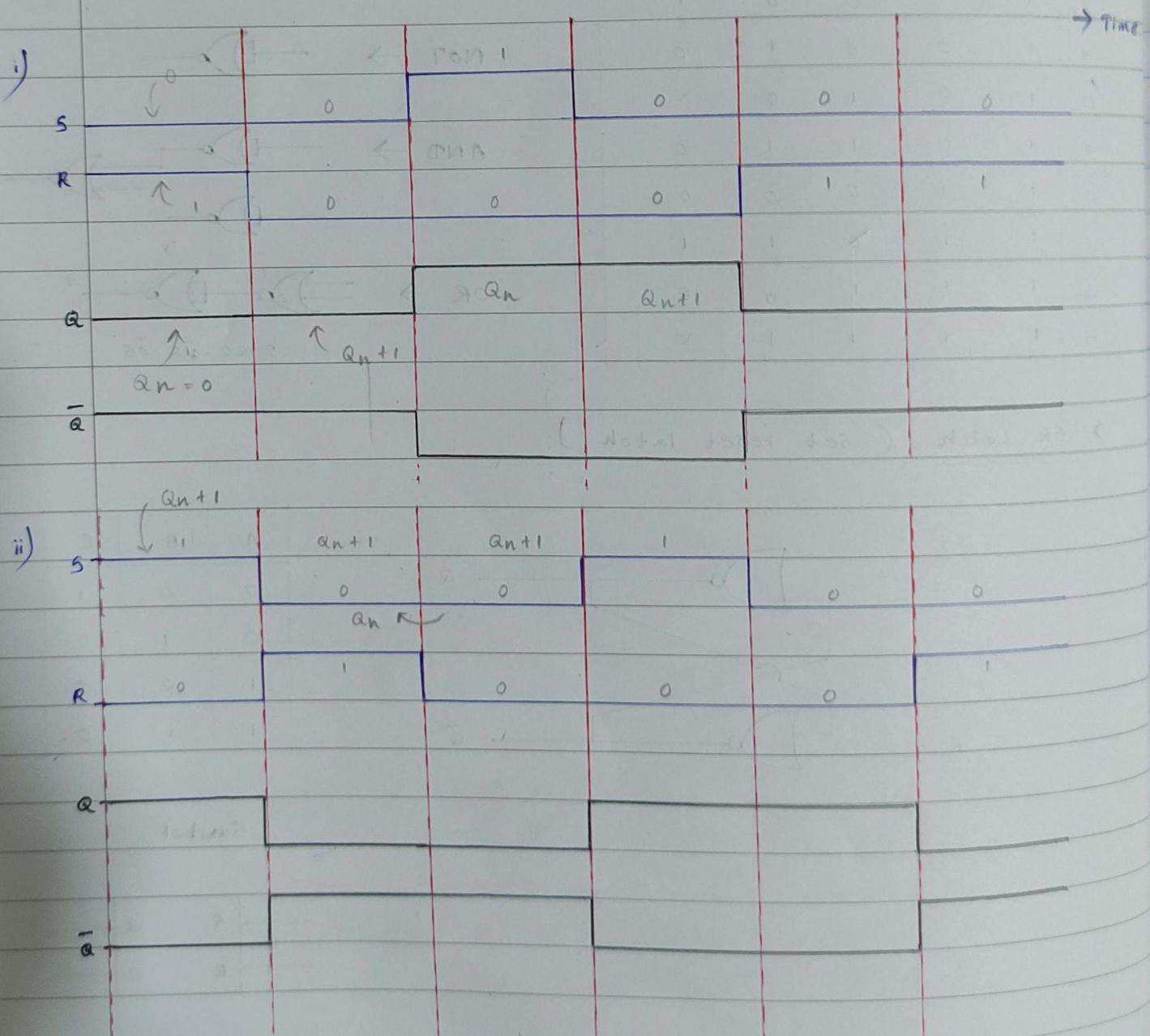
Symbol

S	Q
R	\bar{Q}

Reduced truth table:

	S	R	Q_{n+1}
Hold State	0	0	Q_n
Reset State	0	1	0
Set State	1	0	1
Not Allowed State	1	1	N/A

- Timing diagram of Showing the operation of a SR Latch:



No: _____ Date: _____

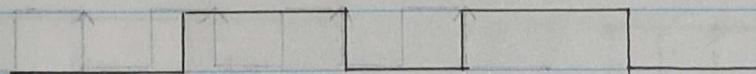
SR Latch Truth table.

S	R	Q_n	\bar{Q}_n	Q_{n+1}	\bar{Q}_{n+1}
0	0	0	1	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	1	not Allowed.	

flip-flops (Triggered latch)

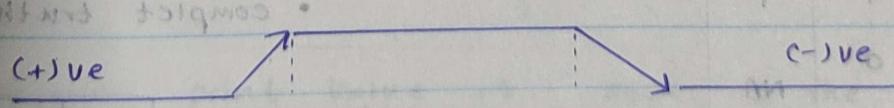
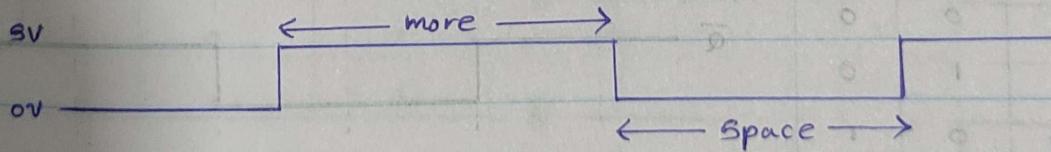
clock

* A clock is a special device that continuously output 0's and 1's. The time it takes for the clock to change from 1 to 0 and back to 1 is called a clock period or clock cycle.

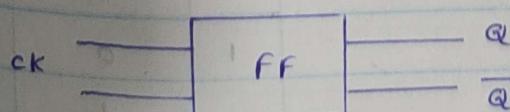


- clock period -

Types of triggering

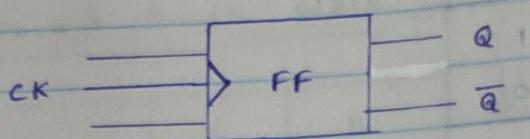


Level Triggering

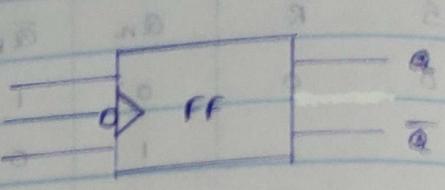


Activated by the mark

Edge Triggering



Activated by the (+)ve



Activated by the (-)ve

SR (Set Reset) Flip-Flop

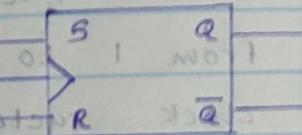
Note

CK	S	R	$Q_n + 1$
0	0	0	Q_n
0	0	1	\bar{Q}_n
0	1	0	Q_n
0	1	1	Q_n
1	1	1	NA

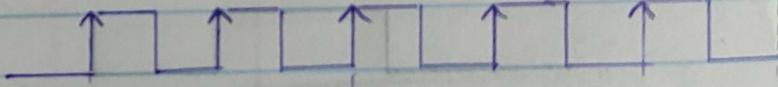
when CK is low

$(Q_n + 1) = Q_n$ (Borrow)

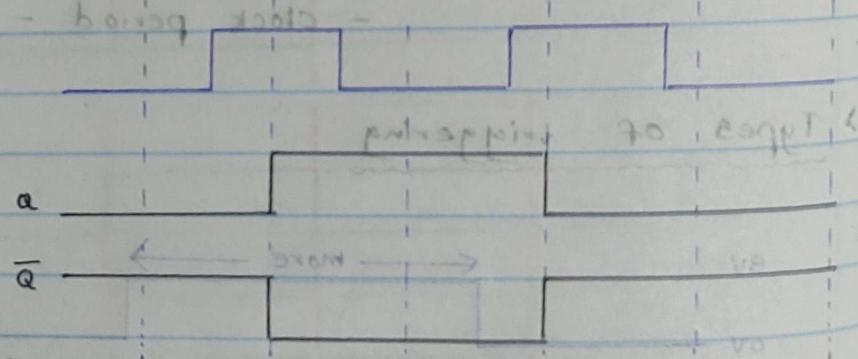
Symbol



Truth table:



S	R	Q_n	Q_{n+1}
0	0	0	0
		1	1
0	1	0	0
		1	0
1	0	0	1
		1	1
1	1	0	NA
		1	



• complete truth table of a

When $CK = 1 \rightarrow$

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

Reduced \rightarrow

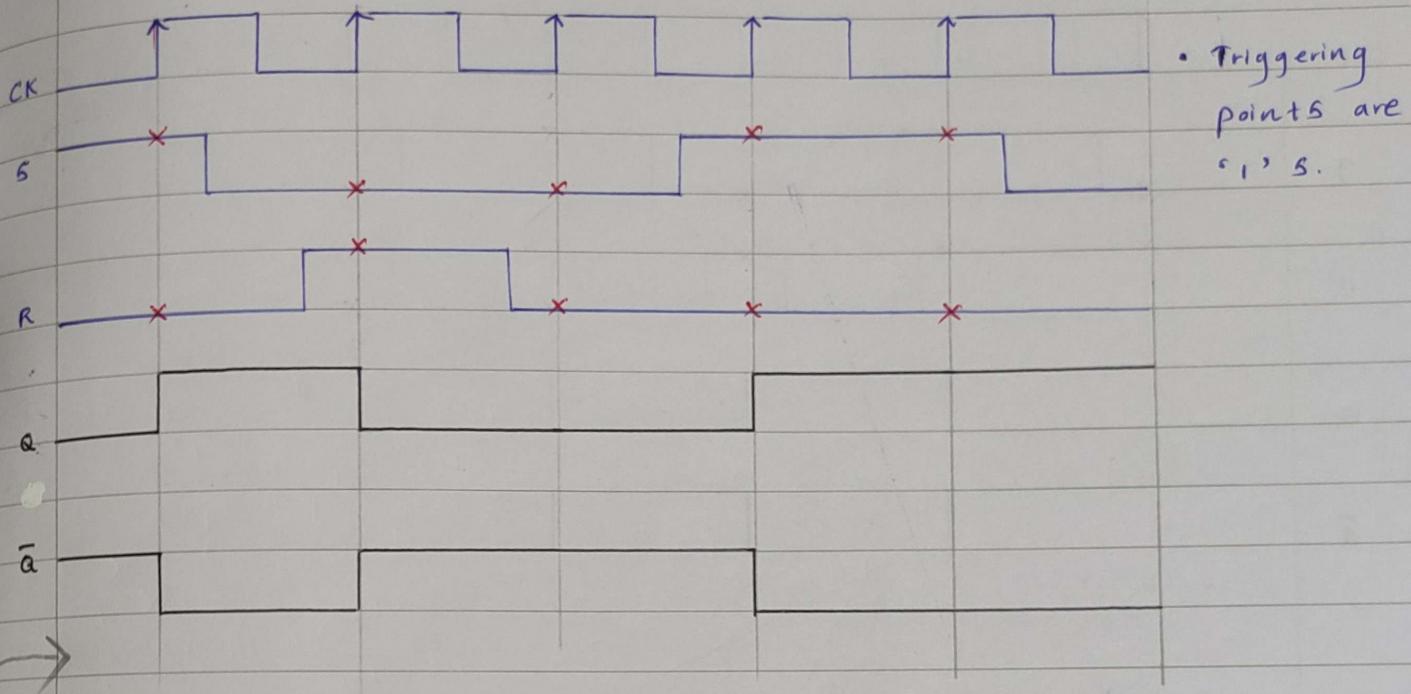
T	Q_n	Q_{n+1}
0	Q_n	
1	\bar{Q}_n	

when
 $CK = 1$

when $CK = 0$

$Q_{n+1} = Q_n$ Atlas

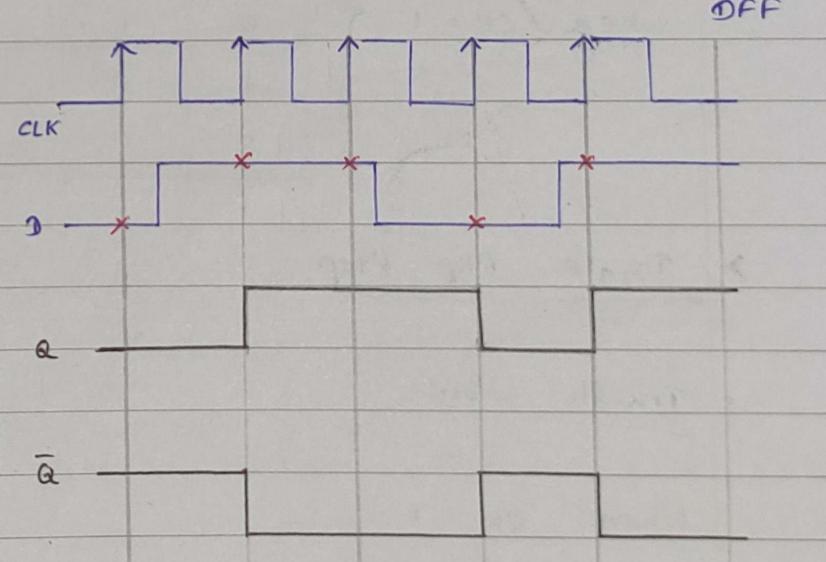
- Draw the output waveform for the (+)ve edge triggered SR.



① flip-flop (Data FF)

Truth table.

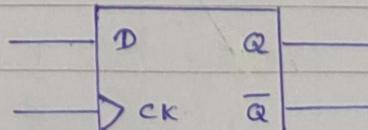
D	Q_n	Q_{n+1}	CLK	Q	\bar{Q}
0	0	0			
	1	0			
1	0	1			
	1	1			



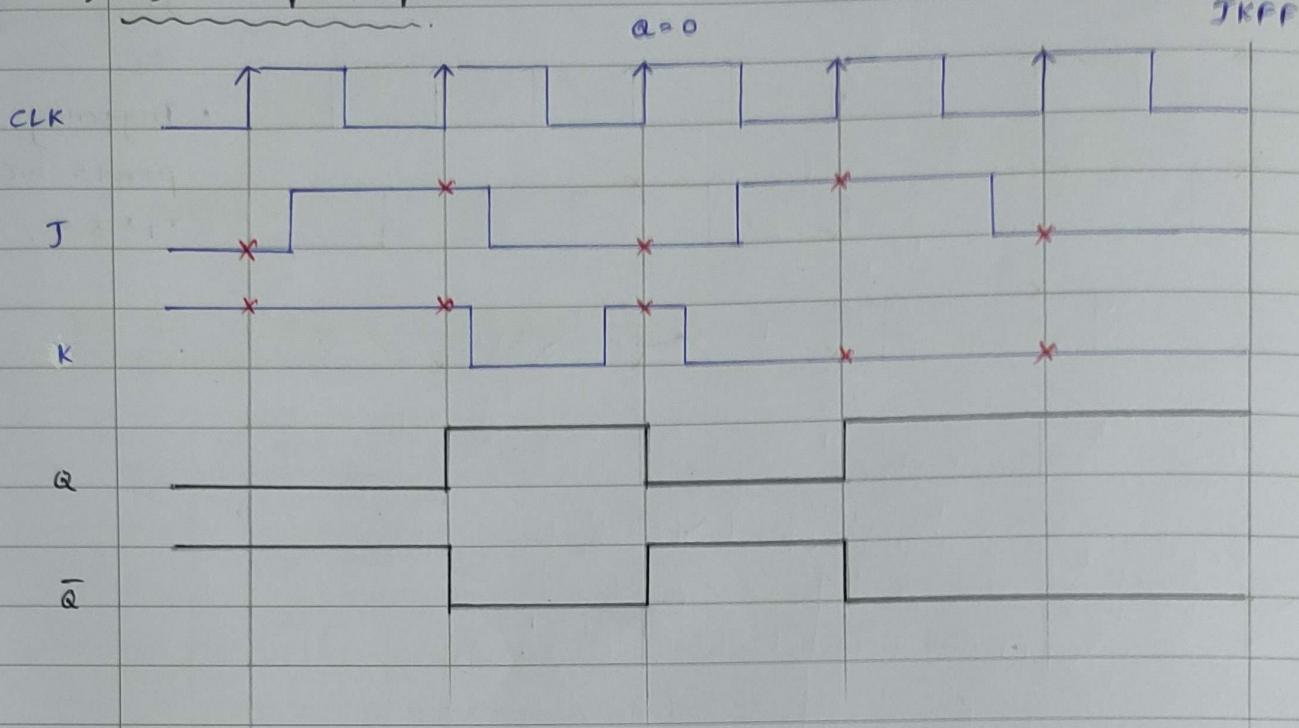
A timing diagram for a D flip-flop. The CLK signal has rising edges at each tick mark. The D input has a transition at the third tick. Red 'x' marks indicate the triggering points. The Q output is high during odd-numbered ticks and low during even-numbered ticks. The \bar{Q} output is the complement of Q. A handwritten note above the diagram says "OFF".

- The Triggering point value is equal to the output (Q) point value.

Symbol,



JK flip-flop



Reduced TT.

when $CK = 0$

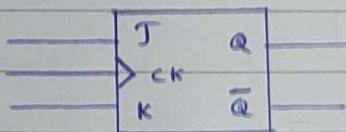
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
0	0	\bar{Q}_n

when $CK = 1$

} when $CK = 0$

} $Q_{n+1} \equiv Q_n$

} Symbol,



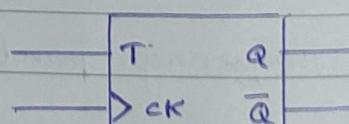
Toggle flip flop

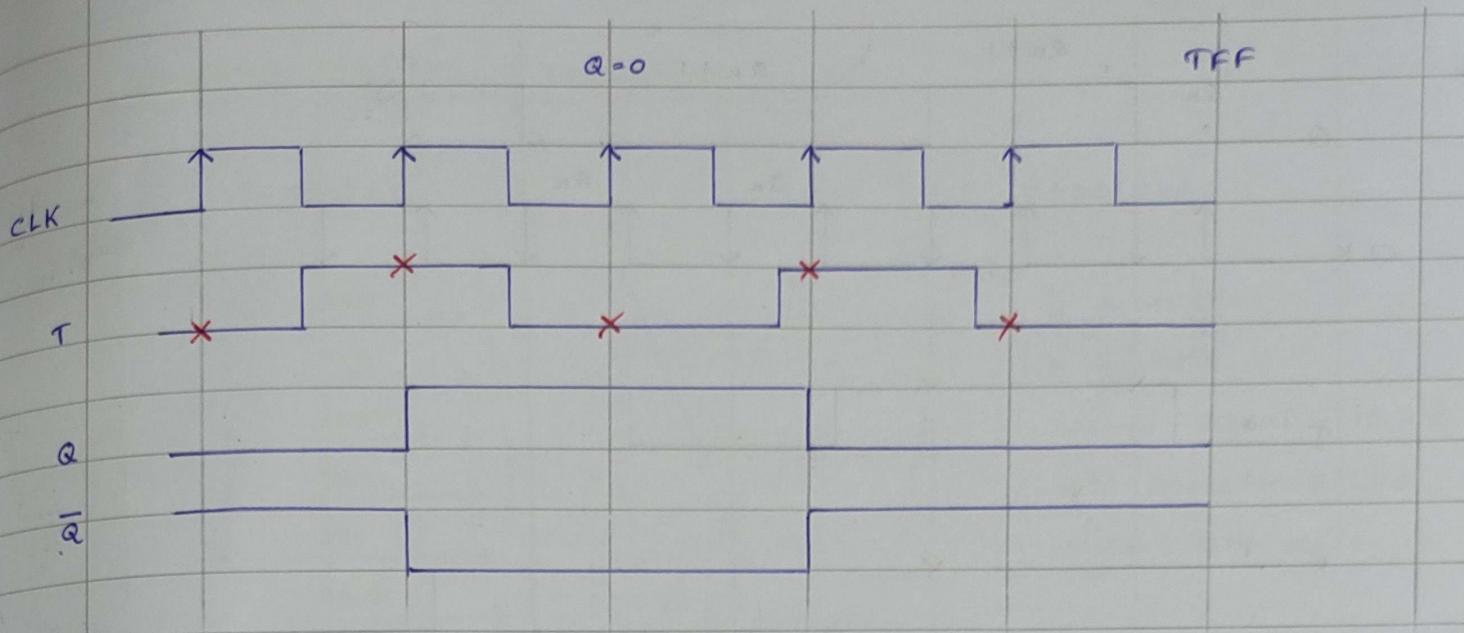
• Truth table.

When $CK = 1$

Symbol,

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





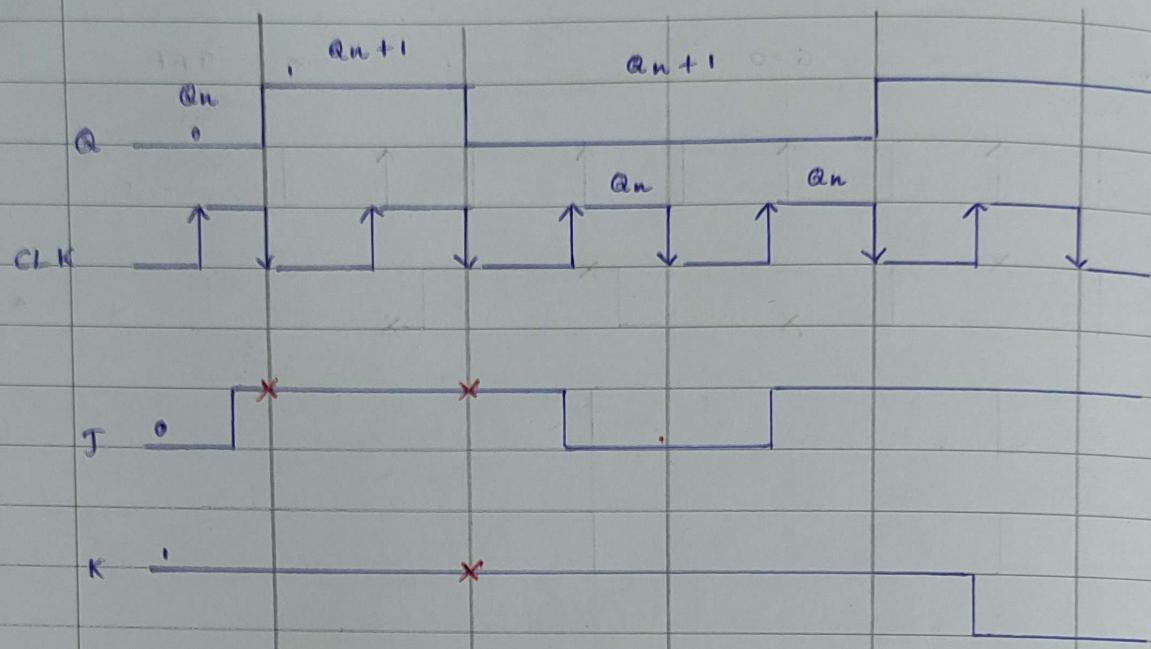
➤ JK flip flop

Truth table :

J	K	Q_n	Q_{n+1}
0	0	0	0
		1	1
0	1	0	0
		1	0
1	0	0	1
		1	1
1	1	0	1
		1	0

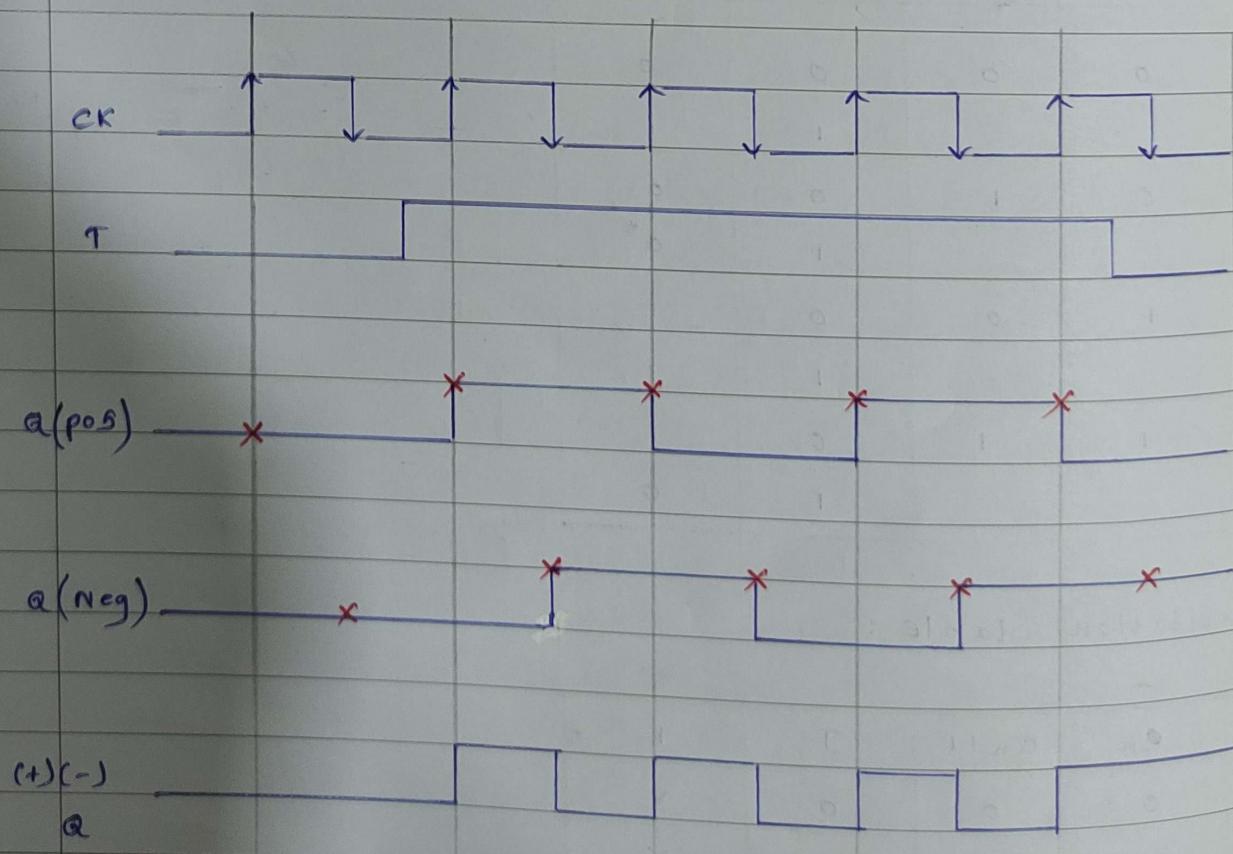
Excitation table :

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



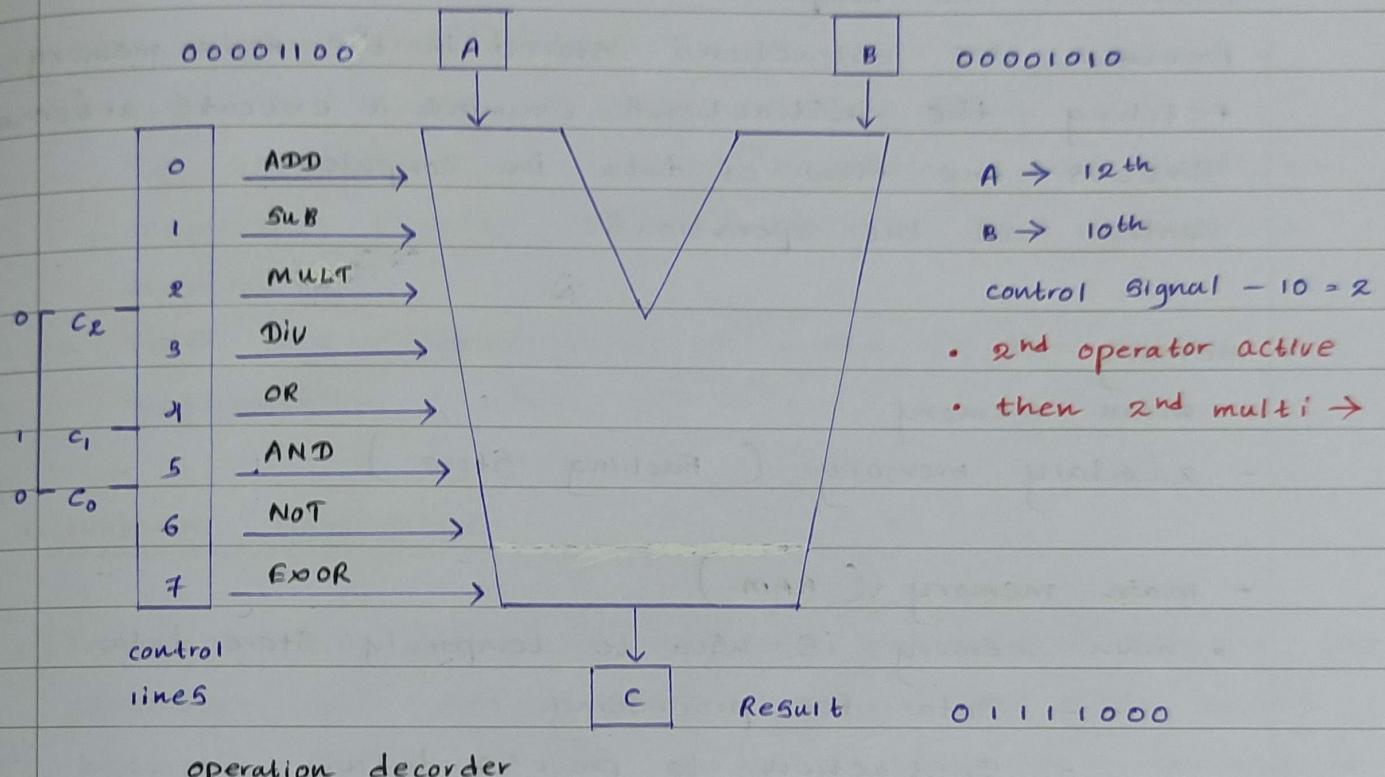
a) Draw the output waveform for a:

- a) (+)ve edge triggered TFF
- b) (-)ve edge triggered TFF
- c) (+) and (-) edge triggered TFF



Ansen should be in step by step 2023.01.03

Computer System Organization



- Arithmetic and logic circuit. - ---> P.g.*
unit

computer systems.

Hardware

Software

* physical parts of a computer system.

* Instructions that tell the hardware what to do

Digital computer

CPU

Main memory

Auxiliary storage

I/O

Devices.

- Central processing unit.

- Brain of the computer
- Execute the instructions stored in the main memory by fetching the instructions, examine & execute after another
- Directs the flow of data in computer
- control all the operations.

- memory

- main memory
- auxiliary memory (Backing Store)
- main memory (RAM)
 - main memory is used to temporarily store data.
 - Data for processing
 - Instruction to process data.
 - Information (processed data) to be sent to an off device or to a Secondary storage.
 - main memory is a volatile memory.
(All the information will be erased after a shutdown - in a Volatile memory)
- Auxiliary Storage (Backing / Secondary Storage)
 - permanent data stores (Non-Volatile)
- Input / output Devices.
 - communication between computer system and outside world.

➤ cpu organizations.• von human architecture.

- Basic of almost all computing done today
- It assumes that every computing pulls data from memory, process it and then send it back to memory.
- This has created what is known as von human bottleneck.

• human bottleneck.

* In the above programs and data are held in memory, the processor and memory are separate, and data moves between the two. In that configuration, latency is unavoidable.

• approaches to overcome the von human bottleneck.1) Caching

- The storage of frequently used data in a special area using RAM, so that is more readable accessible than if it were stored in main memory.

2) prefetching

- moving some data into cache before it is requested to speed access in the event of a request.

3) multithreading

- managing multiple requests simultaneously in separate threads.

4) New types of RAM

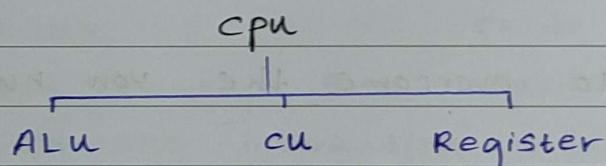
- For example DDR, SDRAM which activates output on both the rising and falling edge of the System clock rather than on just the rising edge, to potentially double output.

5) RAMBUS

- A memory SubSystem consisting of the RAM, the ram controller and the bus (path) connecting RAM to the microprocessor and devices in the computer that use it.

6) processing in memory

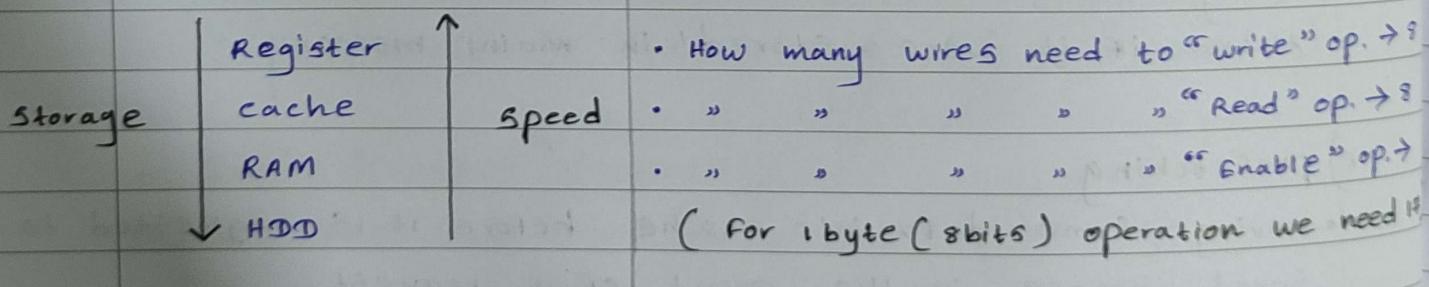
- which integrates a processor and memory in a single microchip.



2023.01.05

Memory Hierachy

8 bit register



* consider about huge amount of data storage, we need huge amount of wires This is a major drawback

Write operation - 1

Read operation - 1

Input / output operation - 1

bit address - 8

> Memory (primary) \rightarrow p.g.*

$$\text{number of addresses} = 2^{\text{Address bus width}}$$

$$\text{capacity of memory} = \text{number of address} \times \text{data bus width}$$

a) calculate the capacity of the memory. if,

$$\text{Address bus width} = 8\text{bits}$$

$$\text{Data bus width} = 8\text{bits}$$

$$\text{no. of address} = 2^8 = 256 \text{ bits}$$

$$\text{capacity of memory} = \text{NA} \times \text{dBW}$$

$$= 256 \times 8\text{bit}$$

$$= \underline{2048 \text{ bits}} \quad \underline{256 \text{ bytes}}$$

a) calculate the capacity of the memory. if,

$$\text{Address bus width} = 16\text{bits}$$

$$\text{Data bus width} = 8\text{bits}$$

$$\text{No. of Address} = 2^{16}$$

$$= 65536 \text{ bits}$$

$$\text{capacity} = 65536 \times 8$$

$$= \underline{524288 \text{ bits}} \quad \underline{65536 \text{ bytes}}$$

Prefixes

$$2^{10} = K$$

$$2^{20} = M$$

$$2^{30} = G$$

$$2^{40} = T$$

a) calculate the capacity of a memory , if,

Address bus width = 32 bit

Data bus width = 8 bit

$$N.A = 2^{32}$$

=

$$\text{capacity} = 2^{32} \times 8 \quad | \quad 2^{30} \times 8 = 2^2 \\ = 1GB$$

a) In a memory there are 1GM address . what is the width of address bus ?

$$1GM = 2^{ABW}$$

$$2^4 \times 2^{20} = 2^{ABW}$$

$$2^{24} = 2^{ABW}$$

$$ABW = 24$$

a) calculate the capacity of the memory

no. of address = 8M

Data bus width = 8bits.

$$\text{capacity of memory} = 8M \times 8\text{bits}$$

$$= 8MB$$

8bit = 1 byte

	$8 \times 64K$	$4 \times 32K$	$2^8 = 2^5 \times 2^{10} = 2^{15}$
DBW	8	4	$2^8 = 2^6 \times 2^{10} = 2^{16}$
ABW	16 bit	15 bit	
capacity	64KB	16KB	

> Address Expansion.

Ex:- construct $8 \times 64K$ memory using $8 \times 32K$ units.

	$8 \times 64K$	$8 \times 32K$
Data bus width	8	8
address bus width	16 bit	15 bit
capacity of memory	64KB	32KB

Ex:- construct $4 \times 32K$ memory using $4 \times 8K$ units.

	$4 \times 32K$	$4 \times 8K$
Data bus width	4	4
address bus w.	15 bit	13 bit
Capacity	16KB	4KB

$$2^x = 32K$$

$$2^x = 32 \times 2^{10}$$

$$x = 2^{15}$$

$$2^x = 8K$$

$$2^x = 8 \times 2^{10}$$

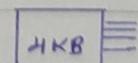
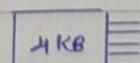
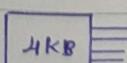
$$x = 13$$

$$\text{Capacity} = \frac{32 \times 4 \times 2^{10}}{8 \times 2^{10}}$$

$$= 16KB$$

$$\text{Capacity} = \frac{8 \times 4 \times 2^{10}}{8 \times 2^{10}}$$

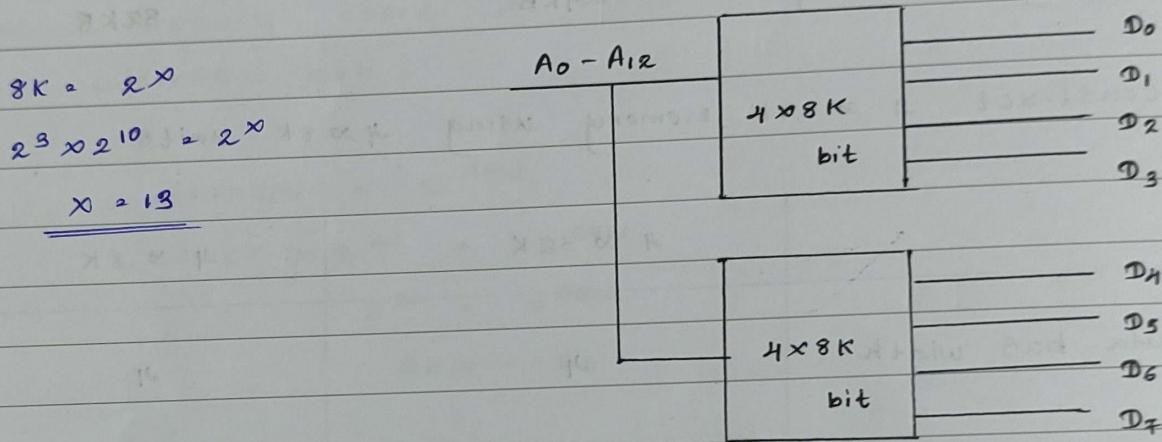
$$= 4KB$$



word expansion

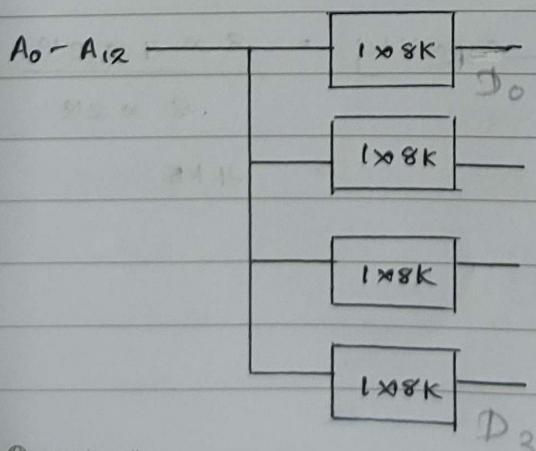
- construct an $8 \times 8K$ bits memory using units of $4 \times 8K$ bit memory.

	$8 \times 8K$	$4 \times 8K$	
DBW	8	8	
ABW	13 bit	13 bit	
memory	8KB	4KB	



- $4 \times 8K$ memory using units of an $1 \times 8K$

	$4 \times 8K$	$1 \times 8K$	$2^X = 8K$
DBW	4	1	$X = 13$
ABW	13	13	
memory	4KB	1KB	

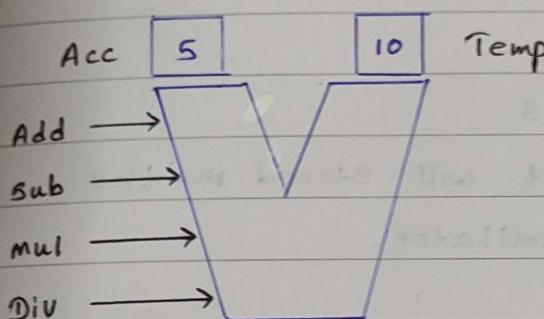


* P.g. <--

Aurithmetic and logical unit.

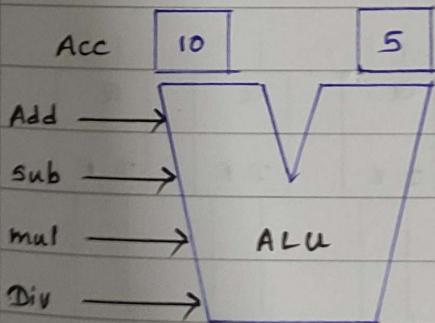
- performs Aurithmetic and logical operations { +, -, ÷, × }
- logical operation make a comparison and act based on the circuit.

ex:- write a program to multiply two numbers A and B. If
 $A = 5$, $B = 10$.



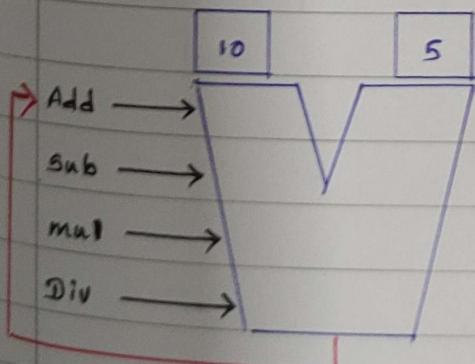
- 1) A and B stored in the accumulator and temp registers respectively.
- 2) control Signal will activate the Multiply operation.
- 3) Result will be Saved in the accumulator.

ex:- you have written a c program to add three numbers A, B and C . IF $A=10$, $B=5$, $C=100$ explain how the ALU will perform this operation.



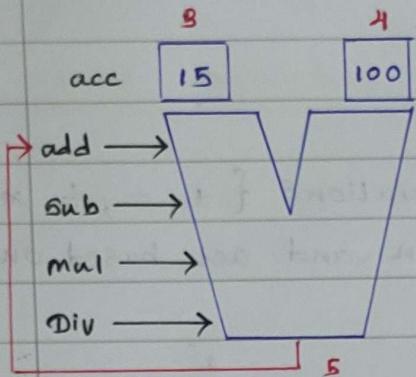
• Step 1

firstly A and B stored in the accumulator and temp registers respectively.



• Step 2

control signal will activate the ADD operation.



• Step 3

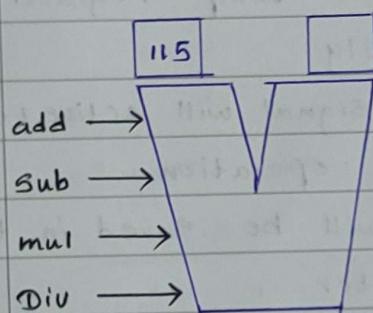
Result will be saved in the accumulator.

• Step 4

c will be saved in the temp register.

• Step 5

Then activate the ADD operation.



• Step 6

Result will be stored in the accumulator.

* p.g. <--

> Memory

8-bit register.

