

P.Y.Q

* VIMP [Fixed question ⇒ 5 OR 6 marks]

i) Conversions →

i) B → D

$$(11010.010)_2 \Rightarrow$$

16 8 4 2 1
| | 0 1 0 . 0 1 0

$$\Rightarrow (26.25)_{10}$$

ii) D → B

$$(26.25)_{10} \Rightarrow$$

$$\begin{array}{r}
 2 | 26 \quad 0 & 0.25 \\
 2 | 13 \quad 1 & \times \frac{2}{2} \quad 0 \\
 2 | 6 \quad 0 \uparrow & \times \frac{0.50}{2} \quad 1 \\
 2 | 3 \quad 1 & \times \frac{1.00}{2} \\
 2 | 1 \quad 1 \\
 0
 \end{array}$$

$$\Rightarrow (11010.01)_2$$

iii) B → O

421 rule

$$(11010.01)_2 \Rightarrow$$

$$\begin{array}{r}
 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \\
 4 \quad 2 \quad 1 \quad 4 \quad 2 \quad 1 \quad 4 \quad 2 \quad 1 \\
 2+1 \quad 2 \quad 2
 \end{array}$$

$$\Rightarrow (32.2)_8$$

iv) B → HD

8421 rule

$$(11010.01)_2 \Rightarrow$$

$$\begin{array}{r}
 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \\
 8 \quad 4 \quad 2 \quad 1 \quad 8 \quad 4 \quad 2 \quad 1 \quad 8 \quad 4 \quad 2 \quad 1 \\
 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1
 \end{array}
 \Rightarrow (1A.4)_{16}$$

v) $D \rightarrow O$

$D \rightarrow \frac{1}{\div \text{ by } 8} \rightarrow \frac{2}{\times \text{ by } 8 \text{ (fraction)}}$

(number)
(number)

$(26.25)_{10}$	\Rightarrow	8 <u>26</u> 2 ↑	$\times \frac{8}{8} \frac{2}{2}$
		8 3 3	
		0	2.00

$$\Rightarrow (32.2)_8$$

vi) $D \rightarrow HD$

$D \rightarrow \frac{1}{\div \text{ by } 16} \rightarrow \frac{2}{(\text{number}) \quad (\text{fraction})}$

$(26.25)_{10}$	\Rightarrow	16 <u>26</u> 10 ↑	$\times \frac{16}{16} \frac{10}{10}$
		16 1 1	
		0	4.00

$$\Rightarrow (1A.4)_{16}$$

vii) $O \rightarrow B$

$$(32.2)_8 \Rightarrow$$

3 Bit

0 32 3 ? . 2 .
equivalent
binary no.

011 010

010

1 = 001

2 = 010

3 = 011

4 = 100

$$\Rightarrow (011010.010)_2$$

viii) $\text{O} \rightarrow \text{HD} \rightarrow \text{B}$

4 Bit
equivalent
Binary no.

$$(1A \cdot 4)_{16} \Rightarrow$$

$$\begin{array}{r} 1 \quad A \quad \cdot \quad 4 \\ 0001 \quad 1010 \quad 0100 \end{array}$$

$$\begin{array}{l} 1 = 0001 \\ 2 = 0010 \\ 3 = 00101 \\ 4 = 0100 \end{array}$$

$$\Rightarrow (00011010 \cdot 0100)_2$$

ix) $\text{O} \rightarrow \text{D}$

$(32.2)_8 \Rightarrow$ Multiply each octal digit by its position weight

$$\begin{array}{r} 3 \quad 2 \quad \cdot \quad 2 \\ \times \quad \times \quad \times \\ 8^1 \quad 8^0 \quad 8^{-1} \end{array}$$

$$24 + 2 \cdot \frac{2}{8} \Rightarrow (26.25)_{10}$$

x) $\text{H.D} \rightarrow \text{O.D}$

$$(1A \cdot 4)_{16} \Rightarrow$$

$$\begin{array}{r} 1 \quad A \quad \cdot \quad 4 \\ \times \quad \times \quad \times \\ 16^1 \quad 16^0 \quad 16^{-1} \end{array}$$

$$16 + A \times 1 \cdot \frac{4}{16} \Rightarrow (26.25)_{10}$$

xi) $\text{O} \rightarrow \text{HD}$

Oct \rightarrow Bin \rightarrow Hex

$$(32.2)_8 \Rightarrow$$

$$\begin{array}{r} 3 \quad 2 \quad \cdot \quad 2 \\ 011 \quad 010 \quad 010 \end{array}$$

$$\Rightarrow (011010 \cdot 010)_2$$

$$\begin{array}{r} 0001 \quad 1010 \quad \cdot \quad 01001 \\ 8421 \quad 8421 \quad \cdot \quad 8421 \end{array}$$

$$\Rightarrow (1A \cdot 4)_{16}$$

xii) $H \cdot D \rightarrow O$

$HD \rightarrow B \rightarrow O$

$$(1A \cdot 4)_{16} \rightarrow$$

$$\begin{array}{r} 1 \quad A \quad . \quad 4 \\ 0001 \quad 1010 \quad . \quad 0100 \end{array}$$

$$\Rightarrow (00011010 \cdot 0100)_2$$

$$000 \ 011 \ 010 \ . \ 010 \ 000$$

421 421 421 421 421

$$\Rightarrow (32 \cdot 2)_8$$

PYQ [Basic \rightarrow should know]

2) Binary operations \rightarrow

i) Addition \rightarrow

A (B _{i+1})	B (B _{i+2})	Sum	Carry	result
0	0	0	0	0
0	01	1	0	1
1	0	1	0	1
1	1	0	1	10

ii) Subtraction \rightarrow

A (B _{i+1})	B (B _{i+2})	Sub	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

iii) Multiplication \rightarrow

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

iv) Division →

$$0 \div 0 = 1$$

$$1 \div 1 = 1$$

P.B. *IMP

3] Complements →

[Question can be ⇒ 5 Marks]

i) By 1's complement →

$$\text{Ex. 1)} \quad (1101)_2 - (0110)_2$$

\downarrow
 $(1001)_2 \rightarrow 1\text{'s complement}$

$$\begin{array}{r} 1101 \\ + 1001 \\ \hline \end{array}$$

$$\underline{= 0110}$$

$$\begin{array}{r} 0110 \\ + 1 \\ \hline \end{array}$$

$$\begin{array}{r} 0111 \\ \hline \end{array}$$

Carry is there, so ans is in positive no.

$$\text{Ex. 2)} \quad (0110)_2 - (1101)_2$$

\downarrow
 $(0010)_2 \rightarrow 1\text{'s complement}$

$$\begin{array}{r} 0110 \\ + 0010 \\ \hline \end{array}$$

$$\begin{array}{r} 1000 \\ \hline \end{array}$$

\downarrow
 $(0111)_2 \rightarrow 1\text{'s complement}$

$$\text{Ans} \rightarrow -(0111)_2$$

∴ Carry is not there, ans is in negative

∴ Take 1's complement again.

ii) 2's complement →

→ Add 1 to 1's complement

$$\text{Ex. } (00110101)_2 \rightarrow$$

$$\begin{array}{r} 11001010 \\ + 1 \\ \hline \end{array}$$

$$(11001011)_2$$

→ 2's complement

iii) Sub by 2's Comp \rightarrow

Ex 1) Ex $\rightarrow (1011)_2 - (0100)_2$

\downarrow
 $(1011)_2 \rightarrow 1\text{'s complement}$

$+ 1100 \rightarrow 2\text{'s complement}$

$$\begin{array}{r} 1011 \\ + 1100 \\ \hline \end{array}$$

$$\begin{array}{r} 10111 \\ + 1100 \\ \hline \end{array}$$

Cary is there, the ans is positive

For final ans, discard the carry,

$\rightarrow (0111)_2$

Ex 2) $(0101)_2 - (1011)_2$

\downarrow
 $(0100)_2 \rightarrow 1\text{'s complement}$

$+ 1$

$0101 \rightarrow 2\text{'s complement}$

$$\begin{array}{r} 0101 \\ + 1011 \\ \hline \end{array}$$

$$10100$$

Cary is not there, find 2's complement of the ans.

$$1010$$

$\hookrightarrow (0101)_2 \rightarrow 1\text{'s complement}$

$+ 1$

$0110 \rightarrow 2\text{'s complement}$

$\Rightarrow (0110)_2$

[No Q till now]

1) Digital Computers →

- 3 Categories → i) Microcomputers
ii) Mini computers
iii) Mainframe computers.

i) Micro Comp →

Constructed to perform specific tasks and constructed with minimum no. of components.
Ex → laptops, desktop computers.

ii) Mini Comp →

Have higher capabilities than micro comp.
appl → business data processing, industrial control, scientific research.

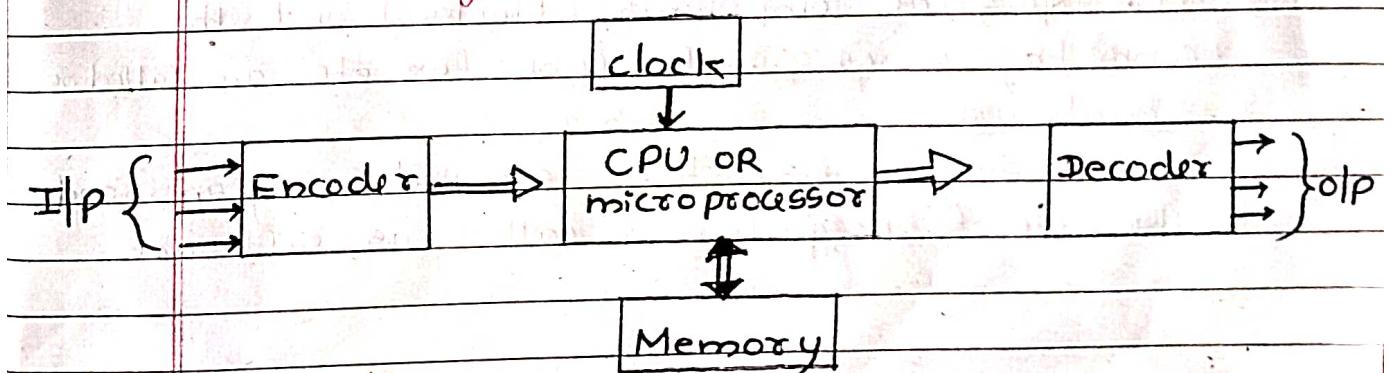
iii) Mainframe Comp → Largest of all types

^{OR} Mainframe Comp. capable of doing storing and processing large amt of data.

appl → military, defence control, business data processing, banks, insurance companies etc.

These comp are extremely fast, also have large memories.

Block diagram of Digital Comp →



i) CPU OR MP →

most imp part of DC. It carries out all computations and decision making. Consists of a block called ALU (Arithmetic and logic Unit), a no. of registers and counters.

ii) clock →

applied to CPU in order to facilitate co-ordinate all operations within the CPU and the Comp.

It acts as a timing signal and it is instrumental for sequencing and synchronizing all operations.

iii) Memory →

Set of instructions is called as program. Such programs entered into the computer using keyboard and stored in part of memory called as program memory.

The CPU then fetches the instructions one by one in the given sequence from the programme memory and executes them.

There is another portion of memory called data memory which is used for storing the info that is being processed by computer.

iv) I/P devices and O/P devices →

Together called as I/O devices.

CPU communicate with outside world through I/O devices.

v) Hardware and software →

The physical components (blocks) and the connecting wires, ports, connectors etc are called as hardware.

A set of instructions is called program and the grip of programs is called as software.

pyo * VIMP [5 OR 6 Marks]

5) Microprocessor →

Arithmetic and logic unit (ALU)

Register array

The MP is a clock driven semiconductor device which consists of electronic logic circuits. MP are manufactured by

Control unit

either large scale integration (LSI) or very LSI techniques.

i) ALU →

This block carries out all computations including arithmetic and logic operations on data. arithmetic operations such as addⁿ, Subs, AND, OR, Inversion, Ex-OR etc.

ii) RA →

This block consists of grp of registers, these reg. are used for temporary storage of data to be processed or the processed data (end result of processing). All these registers are accessible to user.

iii) CU →

This unit produces all the necessary timing and control signals for all the operations.

It controls the exchange of data b/w the micro-processor and memory and I/O devices.

Microprocessor is a sequential digital circuit which consists processes on binary input info and perform different tasks with some executable instruction sequence.

Each mp has a fixed set of instruc. which are coded in binary.

The MP are available in 8 bit, 16 bit, 32 bit, 64 bit categories. The no. of bits shows the size of data which can be processed by the processor.

PYQ *VIMP [5 OR 6 marks]

6) Microcontroller →

	Microprocessor (CPU)				
Memory	T/I O	MC consumes typically relatively less power			
	devices	typically few milliwatts or even microwatts			
	Peripheral devices →				
i) A/D converter					
ii) Timer					
iii) Serial I/O.					

MC is a device that includes MP. Memory, T/I O devices on a single chip using the VLSI technology. This makes it different from microprocessor.

The memory consist of ROM as well as RAM. MC is a complete microcomputer on board.

The MC operates slightly less speed than MP, but their speeds is adequate for the appln. They are used for.

Features →

- i) A MP ranging from 4 to 32 bit processor
- ii) Discrete input and output lines
- iii) Serial input/output ports
- iv) RAM for data storage
- v) clock generator
- vi) Peripherals such as timer, PWM generators, event counters
- vii) A to D converter
- viii) ...

- Appln → MC used in automatically controlled products and devices such as automobile engine control, remote controls, office machines, appliances, toys, traffic control lights etc.

* VIMP [Can Come]

7) Difference b/w MC and MP.

	Microcontroller	Microprocessor
i)	MC has inbuilt RAM or ROM	MP does not have
ii)	Inbuilt timer	Does not have
iii)	I/O Ports are available	not available, requires extra devices like 8250 or 8251.
iv)	Inbuilt Serial port	not available, requires extra devices like 8250 or 8251.
v)	Separate memory is there to store programme and data.	program and data are stored in same memory.
vi)	Many multifunction pins on IC.	less multifunction pins on IC.
vii)	Boolean operation, i.e. operation on individual bit is possible directly.	Boolean operation → is not possible directly
viii)	Few instruction are needed to read/write data to/from ext. memory.	Many instruction are needed to read/write data from/to external memory

P.Y.Q 9) Logic gates \rightarrow VIMP [Q forming topic]
LG are the logic circuits which acts as the basic building blocks of any digital systems. It is an electronic circuit having one or more than one inputs and only one output.

c) classification \rightarrow

i) Basic gates

NOT

AND

OR

IC \rightarrow 7404

ii) Universal (i) iii) Derived

NAND

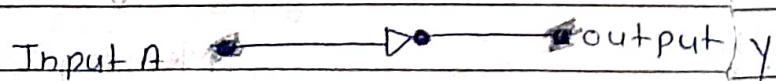
NOR

Ex-OR

Ex-NOR

i) NOT GATE \rightarrow 1 input and 1 output.

Symbol \rightarrow



ii) T. T \rightarrow

A	Y
0	1
1	0

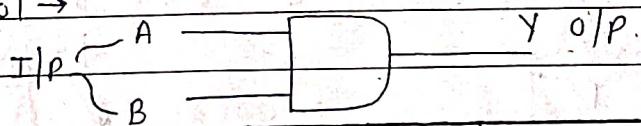
(AQM) chart

3) Exp $\rightarrow y = \bar{A}$ \rightarrow logical inversion.
 IC \rightarrow 7408

ii) AND GATE \rightarrow

2 inputs and 1 output.

i) symbol \rightarrow



ii) T.T \rightarrow

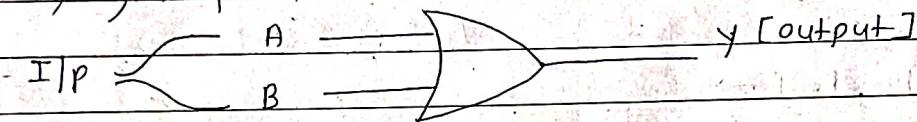
	I/P		O/P	
	A	B	y	
0	0	0	0	
0	0	1	0	
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	1	

iii) B. Exp $\rightarrow y = A \cdot B$ \rightarrow logical multiplication.

IC \rightarrow 7432

iii) OR GATE \rightarrow 2 inputs and 1 output.

i) symbol \rightarrow



ii) T.T \rightarrow

	I/P		O/P	
	A	B	y	
0	0	0	0	
0	0	1	1	
1	0	0	1	
1	0	1	1	
1	1	0	1	
1	1	1	1	

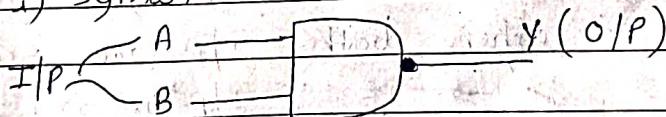
iii) B. Exp \rightarrow

$y = A + B$ \rightarrow logical addition

IC \rightarrow 7400

i) NAND GATE \rightarrow Combinational AND + NOT gate

i) symbol \rightarrow



ii) T.T \rightarrow

I/P		O/P	
A	B	y	
0	0	1	
0	1	1	
1	0	1	
1	1	0	

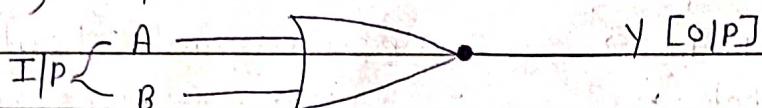
iii) B. Exp \rightarrow $y = \overline{A \cdot B}$

NAND GATE is also called as "Universal gate" bcz we can construct AND, OR and NOT gate using only NAND gates.

IC \rightarrow 7402

i) NOR GATE \rightarrow Comb'n of OR and NOT

i) Symbol \rightarrow



ii) T.T \rightarrow

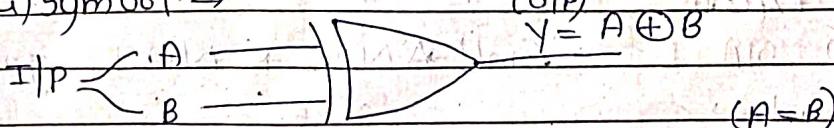
I/P		O/P	
A	B	y	
0	0	1	
0	1	0	
1	0	0	
1	1	0	

iii) B. Exp \rightarrow $y = \overline{A + B}$

IC \rightarrow 7486

i) Ex-OR GATE \rightarrow Two I/P and One o/p.

i) Symbol \rightarrow



ii) F.F \rightarrow

when both I/P are same o/p is low

I/P are not same o/p is high
(A ≠ B)

ii) T.T \rightarrow

I/P		O/P
A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

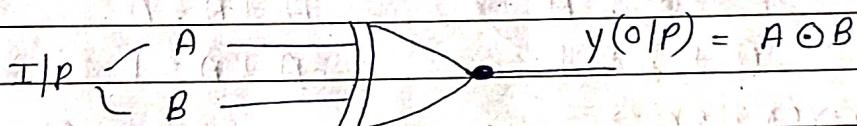
iii) B.Exp \rightarrow

$$y = A \oplus B$$

ii) Ex - ~~IC 7426~~ NOR $\equiv \bar{A}B + A\bar{B}$

iii) Ex - NOR GATE \rightarrow

iv) Sym \rightarrow



when $A = B \rightarrow$ o/p is high

when $A \neq B \rightarrow$ o/p is low.

ii) T.T \rightarrow

I/P		O/P
A	B	y
0	0	1
0	1	0
1	0	0
1	1	1

iii). B.Exp \rightarrow $y = A \oplus B$

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

$$= A \oplus B$$

• Appln of derived gates \rightarrow

i) Ex - OR \rightarrow i) As a magnitude comparator

ii) In Binary to grey code converter

iii) In the adder and subtractor

iv) In parity generator & checker

v) As a controlled subtractor

vi) Ex-NOR \rightarrow i) As even parity generator and checker

ii) As a comparator.

10) Boolean laws \rightarrow :

i) Commutative \rightarrow

$$A + B = B + A$$

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

ii) Associative \rightarrow

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

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

iii) AND law \rightarrow

$$A \cdot 0 = 0 \quad | \quad A \cdot A = A$$

$$A \cdot 1 = A \quad | \quad A \cdot \bar{A} = 0$$

iv) OR \rightarrow

$$A + 0 = A \quad | \quad A + A = A$$

$$A + 1 = 1 \quad | \quad A + \bar{A} = 1$$

v) Inversion \rightarrow

$$\bar{\bar{A}} = A$$

vi) Distributive \rightarrow

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

vii) Other \rightarrow

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

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

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

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

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

PYQ's *VIMP \rightarrow Fixed Question (5 or 6) marks)

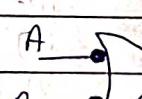
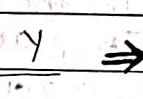
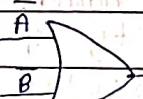
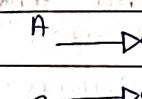
i) De-morgan's theorem \rightarrow

ii) Theorem \rightarrow

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

NAND = Bubbled OR

iii) Symbol \rightarrow



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

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

ii) T. T \rightarrow

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

from this \rightarrow

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

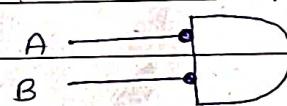
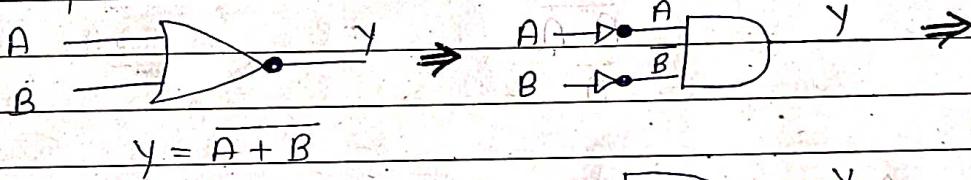
ii)

Theorem 2 \rightarrow

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

NOR = Bubbled AND

ii) Symbol \rightarrow



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

ii) T. T \rightarrow

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

From this,

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

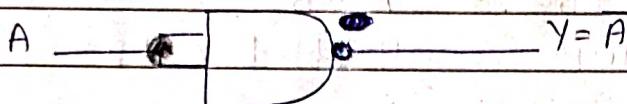
ii) Universal gates \rightarrow

NAND and NOR gates \in are called as Universal gates becz with the help of these gates we can implement any boolean exp.

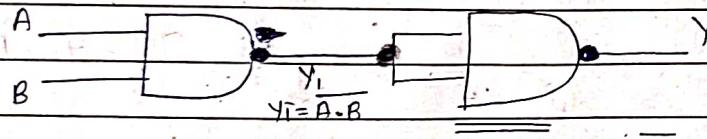
we can built any logical circuit with the help of only NAND gate or only NOR gate.

i) NAND gate only \rightarrow i) NOT using NAND \rightarrow

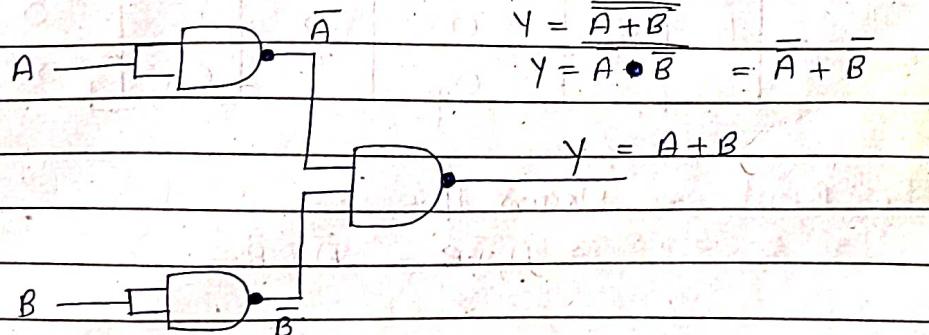
$$\text{Input} = A - B = A$$



$$\begin{aligned} Y &= \overline{A \cdot B} \\ &= \overline{A \cdot A} \\ &= \overline{A} \end{aligned}$$

2) AND \rightarrow 

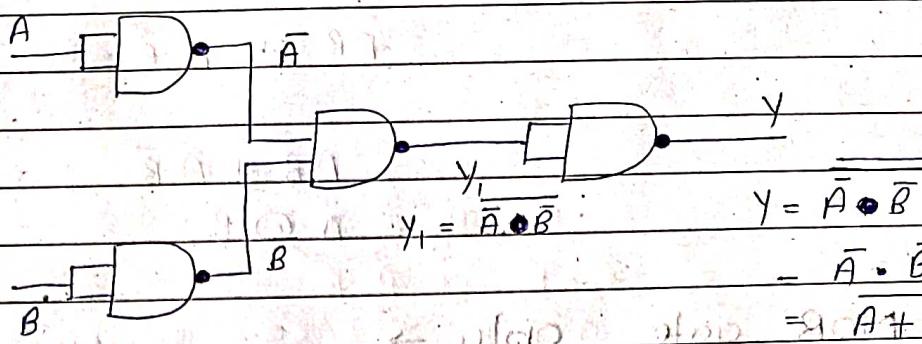
$$\begin{aligned} Y &= \overline{A \cdot B} = \overline{Y_1} \\ &= A \cdot B \end{aligned}$$

3) OR \rightarrow 

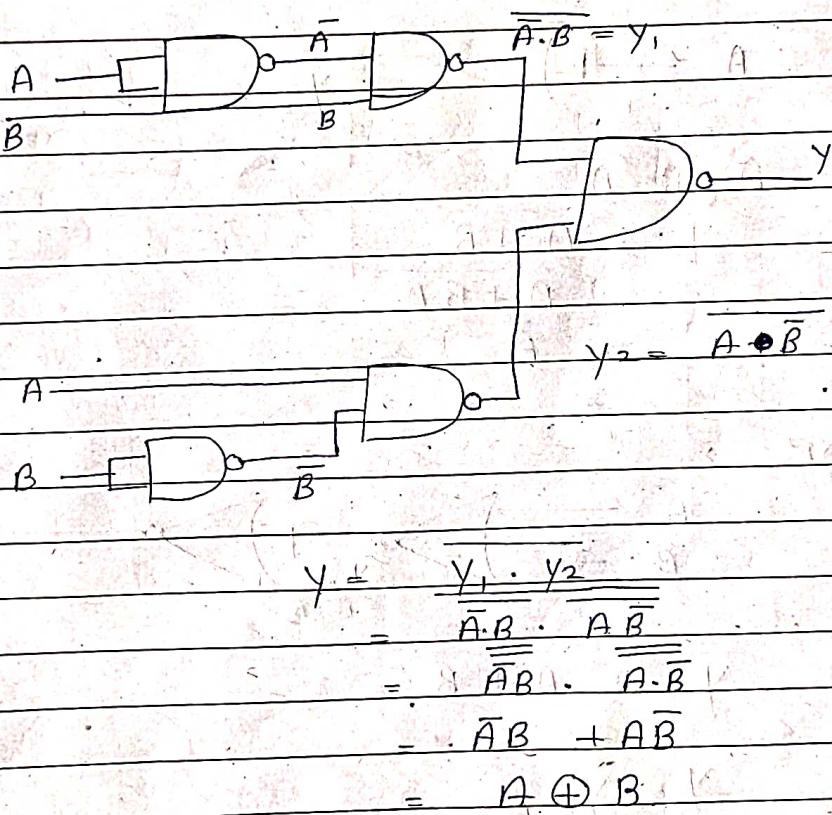
$$\begin{aligned} Y &= \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A} + \overline{B}} \\ &= A + B \end{aligned}$$

4) NOR \rightarrow

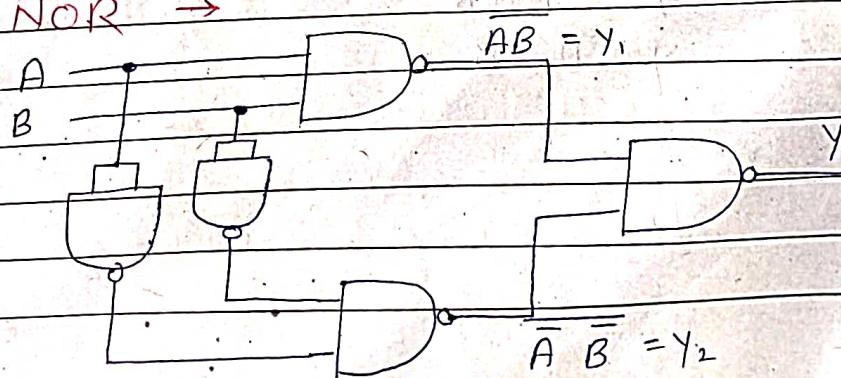
$$\begin{aligned} y &= \overline{A + B} \\ &= \overline{A} \cdot \overline{B} \\ &= \overline{\overline{A} \cdot \overline{B}} \end{aligned}$$



5) Ex - OR \rightarrow



6) Ex - NOR \rightarrow



$$Y = \overline{Y_1 \cdot Y_2}$$

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

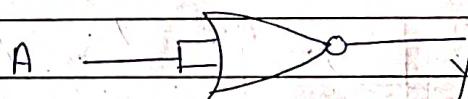
$$= \overline{AB} + \overline{\overline{A}\overline{B}}$$

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

$$= A \odot B$$

ii) NOR gate only \rightarrow

i) NOT \rightarrow



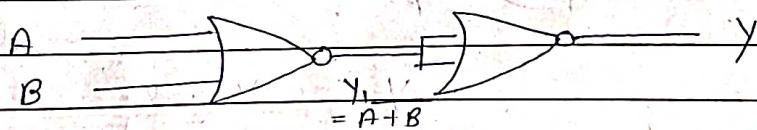
$$A = B = A$$

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

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

$$= \overline{A}$$

2) OR \rightarrow



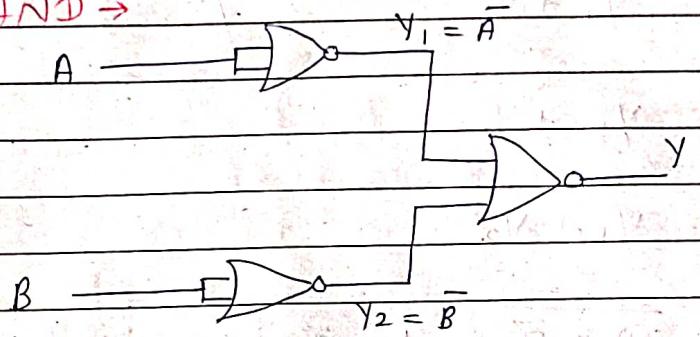
$$y_1 = \overline{A + B}$$

$$y = \overline{y_1}$$

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

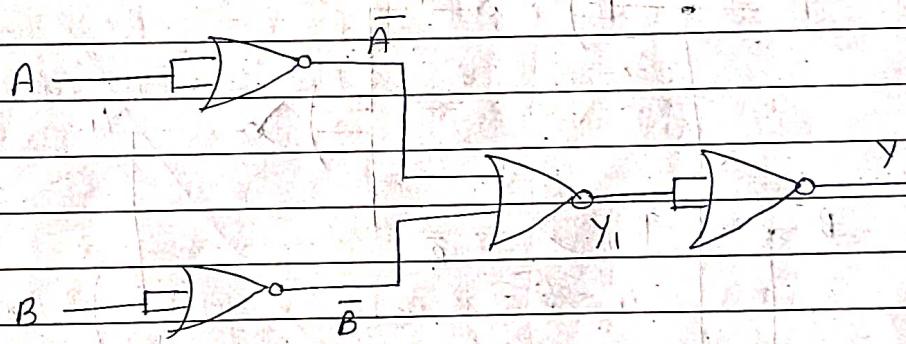
$$y = A + \overline{B}$$

3) AND \rightarrow



$$\begin{aligned}
 y &= \overline{y_1 + y_2} \\
 &= \overline{\bar{A} + \bar{B}} \\
 &= \bar{A} \cdot \bar{B} \\
 &= A \cdot B
 \end{aligned}$$

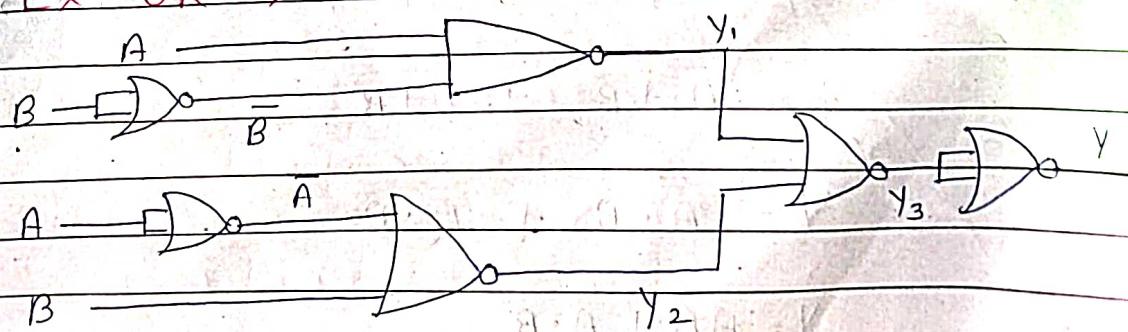
4) NAND \rightarrow



$$\begin{aligned}
 y_1 &= \bar{\bar{A} + \bar{B}} \\
 y &= \overline{\bar{A} + \bar{B}} \\
 y &= \overline{\bar{A} + \bar{B}}
 \end{aligned}$$

$$\begin{aligned}
 y &= \overline{\bar{A} + \bar{B}} \\
 &= \bar{A} \cdot \bar{B}
 \end{aligned}$$

5) Ex - OR \rightarrow



$$y_3 = \overline{y_1 + y_2} \Rightarrow y = \overline{\overline{y_1 + y_2}}$$

$$y = \overline{\overline{y_1 + y_2}}$$

$$y = y_1 + y_2$$

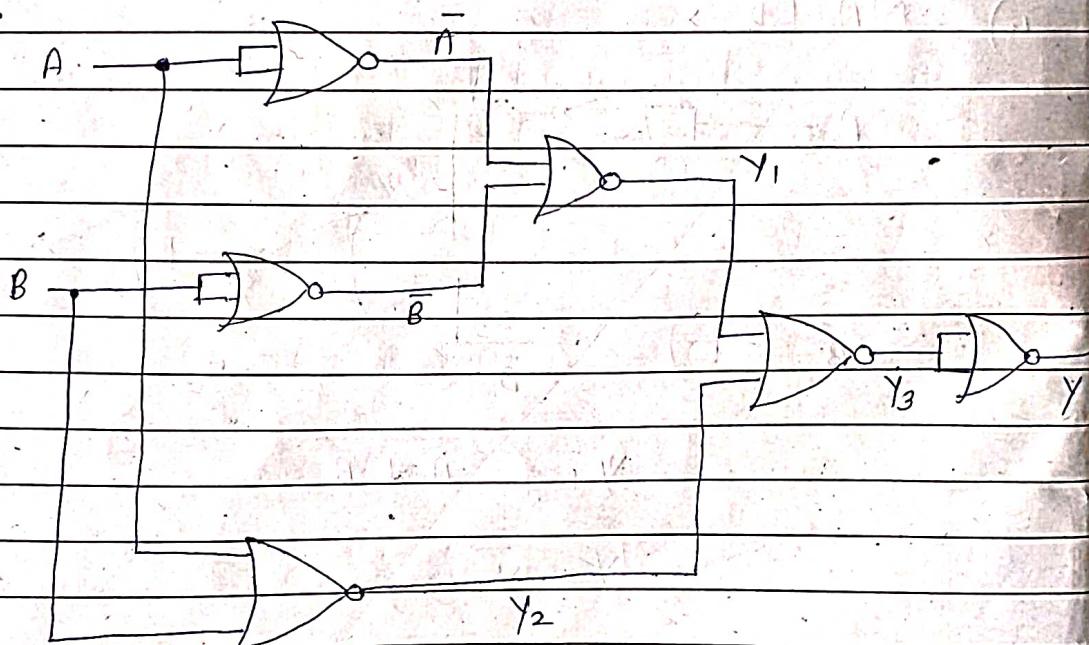
$$y = \overline{A + \bar{B}} + \overline{\bar{A} + B}$$

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

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

$$= A \oplus B$$

c) Ex-NOR \rightarrow



$$y_1 = \overline{\bar{A} + \bar{B}} \quad y_2 = \overline{A + B}$$

$$y_3 = A \overline{y_1 + y_2}$$

$$y = \overline{\overline{y_3}} = \overline{\overline{y_1 + y_2}} = y_1 + y_2$$

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

$$y = \bar{A} \cdot \bar{B} + \bar{A} \cdot B$$

$$y = AB + \bar{A} \cdot \bar{B}$$

$$y = A \odot B$$

Ques → * VIMP (\oplus can come)

13) Adders →

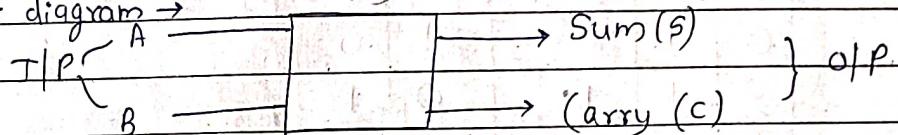
Binary adders of two types →

1) Half →

combinational logic circuit

⇒ 2 I/P and 2 O/P.

a) Block diagram →



b) T-T →

I/P		O/P	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

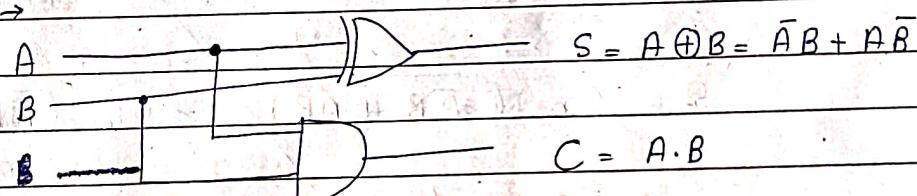
c) Equations →

$$S = A \oplus B$$

$$S = \bar{A}B + A\bar{B} \quad (\text{Ex-OR gate})$$

$$C = A \cdot B \quad (\text{AND gate})$$

d) Sym →



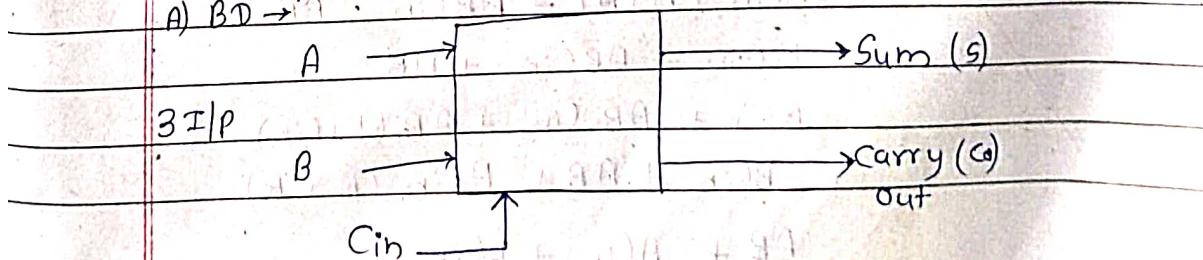
• Disadvantage → It can perform 1-bit addn only.

∴ addn of 3 bits is not possible to perform by using half adder. Hence we can not use half adder in practical.

2) Full →

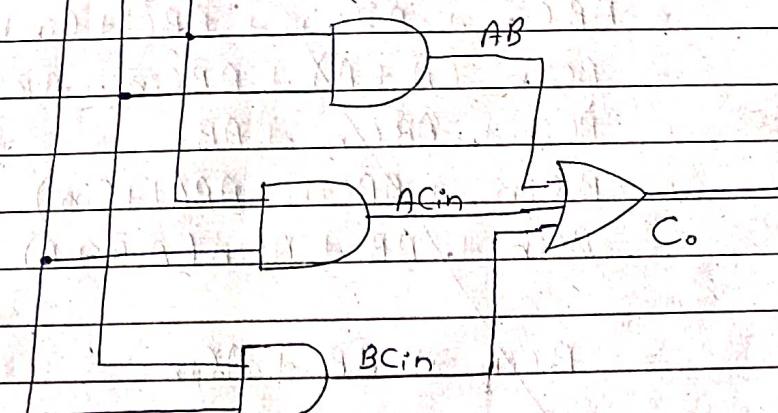
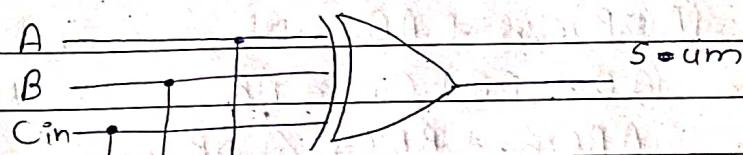
This is 3 I/P 2 O/P. Combinational logic circuit.

a) BD →

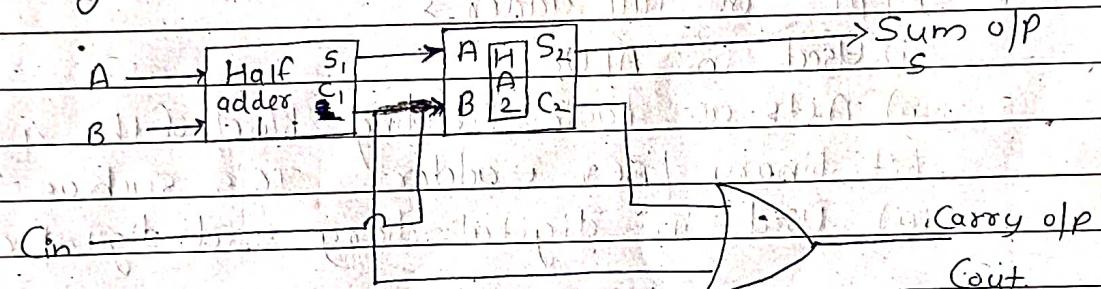


B) $T \rightarrow T$

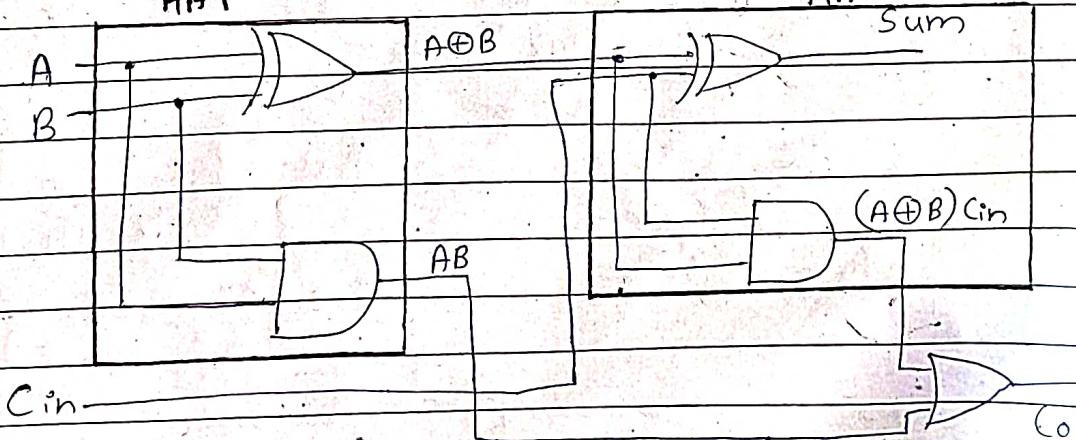
I/P			O/P	
A	B	Cin	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	•	1

d) Gym \rightarrow 2) Full using half \rightarrow

Full adder can be constructed by using two half adders.

Symbol (circuit) \rightarrow

HA1



$$S = A \oplus B \oplus C_{in}$$

Both the Sum and Carry of full adder and full adder using half is same.

∴ Eu

• Appl'n of full adder →

- i) Used in ALU.
- ii) Acts as basic building block of the 4 bit/8 bit binary / BCD adder IC's such as 7483.
- iii) Used in digital diary and digital computers.