

Venilog, HDL, JK flip-flop as a venilog module,
4-bit Synchronous Counter.

3-to-8 decoder, de-multiplexer, implement an 8-to-
1 multiplexer, using 2-to-1 multiplexer, distinguish
Combinational and sequential logic circuit.

2021

1.a) weighted and non weighted codes are used
to represent the decimal numbers.

Weighted codes: In weighted codes, each digit is
assigned a specific weight according to its
position. For example, in 8421 BCD code, 1001
the weights of 1, 0, 0, 1 (from left to Right) are
8, 4, 2, 1 respectively.

Non-weighted codes The non-weighted codes are
not positionally weighted. In other words, each
digit position within the number is not assigned
a fixed value (or weight).

Excess-3 and Gray code are non-weighted
codes.

Unit-distance code: It is an unweighted code
that changes at only one digit position when
going from one number to the next in a
consecutive sequence of numbers.
Example: Gray Code.

1. v) Given, $A = (00101101)_2$
 $B = (11011010)_2$

2.c) A parity bit is a check bit, which is added to a block of data for error detection purpose. The value of the parity bit is assigned either 0 or 1 that makes the number of 1s in the message block either even or odd depending upon the type of parity. Parity check is suitable for single bit error detection only.

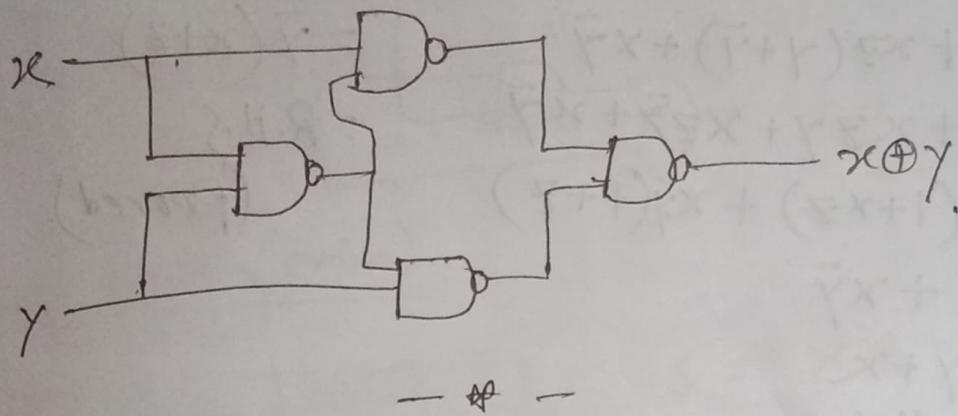
Two types of parity bit checking are,

- i) Even parity - Here the total number of bits in the message is made even.
- ii) Odd parity - Here the total number of bits in the message is made odd.

2021
2.a)

Draw a circuit diagram to implement an Exclusive-OR gate using four NAND gate.

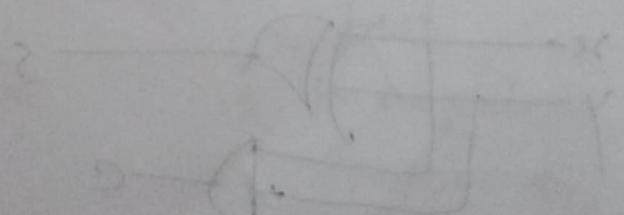
$$X\text{-OR gate} = X \oplus Y = \bar{x}\bar{y} + x\bar{y}$$



- * -

2.b) Design a circuit diagram for converting BCD code into 2-out-of-5 code.

BCD	2-out-of-5
0000	000
0001	001
0010	010
0011	011
0100	100
0101	101
0110	110
0111	111
1000	-



$$2.(c) i) Y + x\bar{y}\bar{z} + xz = x + y$$

$$\text{L.H.S, } Y + x\bar{y}\bar{z} + xz$$

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

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

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

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

$$= Y + xzY + xz\bar{y} + x\bar{y}$$

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

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

$$= Y + x$$

$$= \text{R.H.S}$$

(Proved)

— * —

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

L.H.S

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

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

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

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

= R.H.S

(Proved)

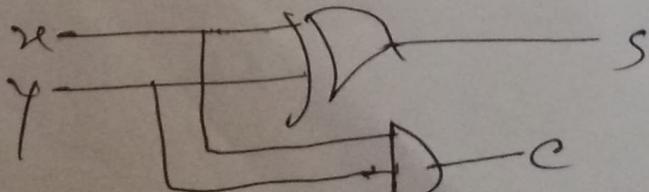
3. a) i) write down the truth table for a half adder.
expression of half adder:

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

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

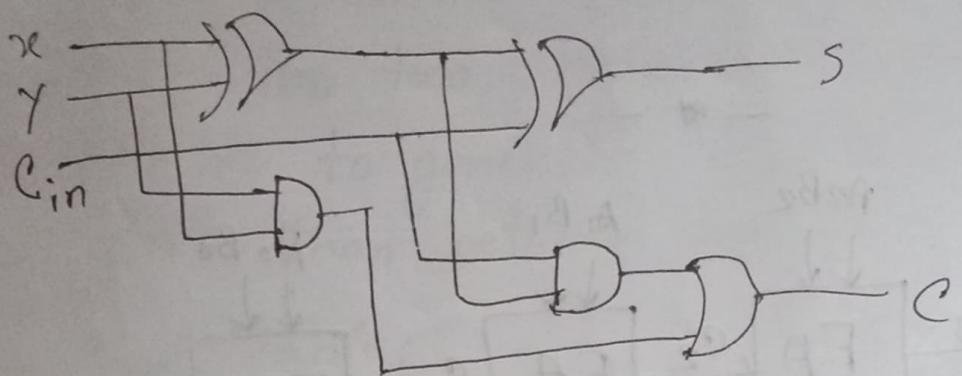
$$C = xy$$

- ii) Derive the logic circuit from the truth table of a half adder.



— * —

iii) Draw a full adder circuit by using two half adders.



Truth table (extra)

x	y	Cin	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$\begin{aligned}
 S &= x \oplus y \oplus \text{Cin} \\
 C &= \text{Cin}(\bar{x}\bar{y} + x\bar{y}) + xy \\
 &= \text{Cin } x\bar{y} + \text{Cin } \bar{x}y + xy
 \end{aligned}$$

- * -

b) Difference between a parallel adder and a serial adder?

parallel adder

serial adder

- | | |
|--|----------------------------------|
| 1. Parallel adder is generally faster. | 1. Serial adder is less fast. |
| 2. It requires more components. | 2. It requires fewer components. |

3. All the bits are added simultaneously.

3. Addition is performed bit-by-bits starting from the LSB.

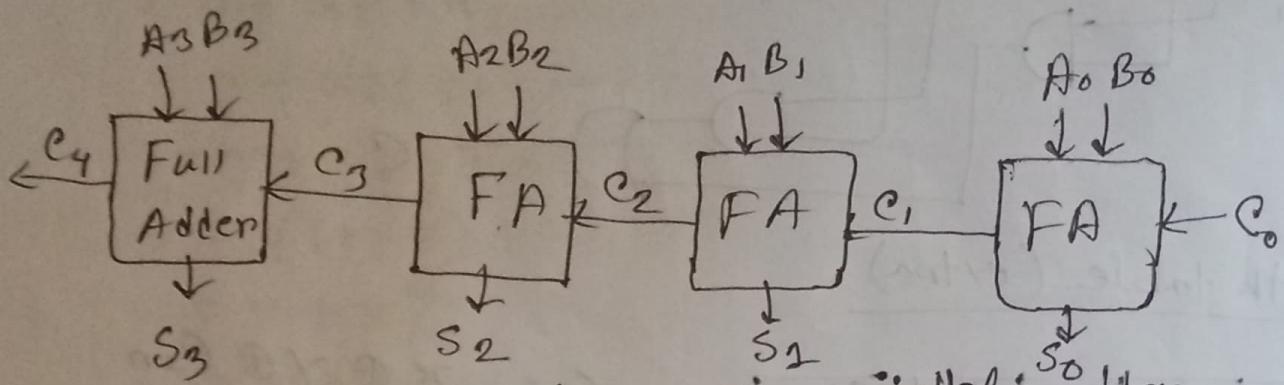
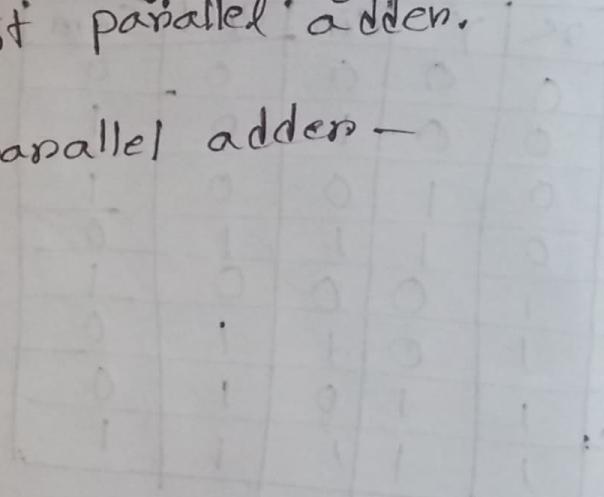


Diagram of 4-bit parallel adder.

Timing diagram of 4-bit parallel adder -



2021
 3.c) 1-bit Magnitude Comparators: A Comparator used to compare two bits is called a single-bit Comparator. It consists of two inputs each for two single-bit numbers and three outputs to generate less than, equal to and greater than between two binary numbers.

Truth table:

A	B	$A \leq B$	$A = B$	$A > B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

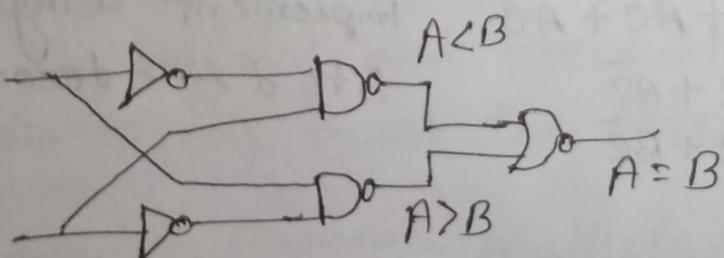
logical expressions

$$A > B \Leftrightarrow AB'$$

$$A \leq B \Leftrightarrow A'B$$

$$A = B \Leftrightarrow A'B' + AB$$

Logic circuit:



denire the formula we can,

$$((A < B) + (A > B))' = (A = B)$$

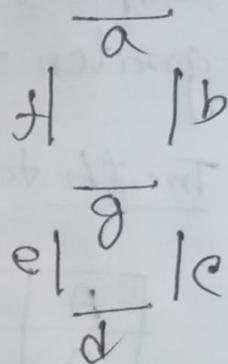
- * -

2021

4(a) BCD to 7-segment decoder

7-Segment
display

Digit	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	1
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	1	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0



- At -

Q. b) Given, $Y_1 = A\bar{B} + AC + \bar{A}\bar{C}$ implement using 74LS138
 $Y_2 = \bar{A}C + A\bar{C}$ 3 to 8 line decoder.
 $Y_3 = \bar{B}\bar{C} + B\bar{C}$

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

Q.e) write some typical applications of multiplexer and demultiplexer.

Applications of Multiplexer:

Multiplexers are used in various applications wherein multiple-data need to be transmitted by using a single line.

- i) Communication System: A Communication System has both a Communication network and a transmission system. The efficiency of Communication System can be increased considerably using multiplexer.
- ii) Telephone Network: In telephone network, multiple audio signals are integrated on a single line for transmission with the help of multiplexers. In this way, multiple audio signals can be isolated and eventually, the desire audio signals reach the intended recipients.
- iii) Computer Memory- Multiplexers are used to implement huge amount of memory into the computer, at the same time reduces the number of copper lines required to connect the memory to other parts of the computer circuit.

Applications of Demultiplexer :-

- i) Demultiplexer is used to connect a single source to multiple destinations. The main application area of demultiplexers is communication system, where multiplexers are used. Most of the communication system are bidirectional, they function in both ways (transmitting and receiving signals).
- ii) Communication System - Communication System use multiplexers to carry multiple data like audio, video and other form of data using a single line for transmission easier. This process make the transmission easier.
- iii) ALU (Arithmetic Logic Unit) - In an ALU circuit, the output of ALU can be stored in multiple registers or storage units with the help of demultiplexers. The output of ALU is fed as the data input to the demultiplexer.
- iv) Serial to parallel converter - A serial to parallel converter is used for reconstructing parallel data from incoming serial data stream. In this technique, serial data stream is given as data input to the de-multiplexer at the regular intervals.

4.c) The multiplexers are commonly used in communication systems, telephone networks, computer memories, etc.

The demultiplexers are used in communication systems, reconstruction of parallel data, ALU, etc.

2020

2.b) 93_{10}

$$\begin{array}{r} 2 | 93 \\ \underline{2} | 48 \rightarrow 1 \\ \underline{2} | 23 \rightarrow 0 \\ \underline{2} | 11 \rightarrow 1 \\ \underline{2} | 5 \rightarrow 1 \\ \underline{2} | 2 \rightarrow 1 \\ \underline{2} | 1 \rightarrow 0 \\ 0 \rightarrow 1 \end{array}$$

$$= (1011101)_2$$

} The base 2 number's actual length, in bits: 7

A signed binary's bits length must be equal to a power of 2, as of:

$$2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32, 2^6 = 64$$

$$\therefore \text{Signed binary: } 93_{10} = 01011101$$

positive

$$-93_{10} \text{ Signed binary} = 11011101$$

$$93_{10} \text{ 1's Complement} = 10100010$$

$$-93_{10} \text{ " } \quad \quad \quad 2^6 00100010$$

$$2's \text{ complement of } 93_{10} = 10100011$$

$$2's \text{ complement of } -93_{10} = 00100011$$

→ At →

2(c) hamming code

	1	0	1	1	0	0	1	0	1	1	0	
P ₁	P ₁			1		1	1	1	1	1	1	
P ₂		P ₂	1			1	1	1				
P ₄			P ₄	1	1	1						
P ₈							P ₈	1	1	1		

$$\begin{aligned} P_1 &= 110110 = 0 \\ P_2 &= 010110 = 1 \\ P_4 &= 100110 = 0 \\ P_8 &= 0110 = 0 \end{aligned}$$

$$(P_8 P_4 P_2 P_1) = (0010)_2 = 2_{10} \text{ position}$$

$$\text{Correction code : } (10110010100)_2$$

$$2.c) F(A, B, C, D) = \sum (1, 3, 7, 8, 11, 15) + d\sum (0, 2, 5, 9)$$

AB	CD	00	01	11	10
00	X	1	1	X	
01	X	1	1		
11				1	
10	(1)	X	1		

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

→ At →

2018

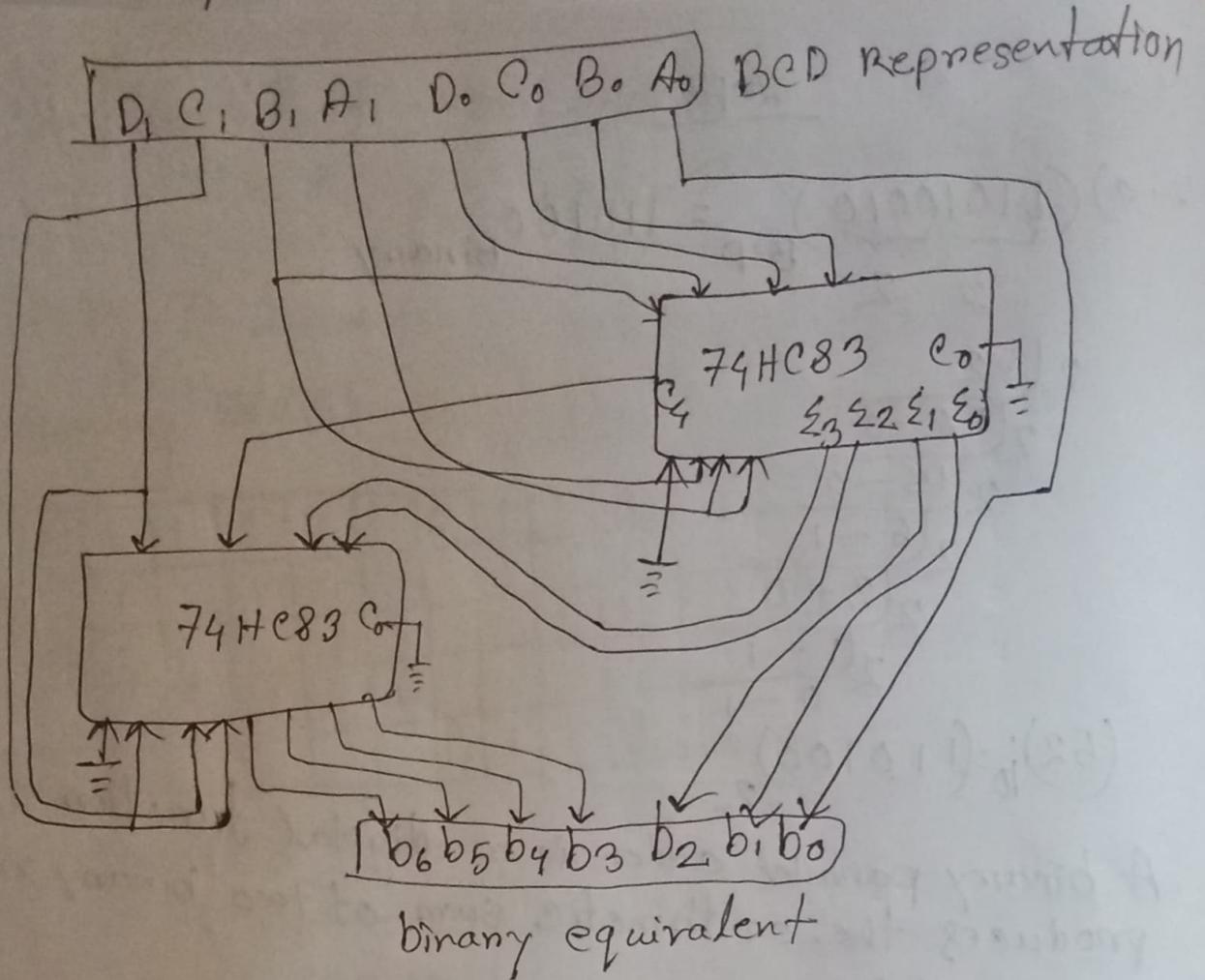
$$2. a) (\underbrace{01010010}_{\substack{5 \quad 2}})_{BCD} = 110100 \text{ Binary}$$

$$\begin{array}{r} 2 | 52 \\ 2 | 26 - 0 \\ 2 | 13 - 0 \\ 2 | 6 - 1 \\ 2 | 3 - 0 \\ 2 | 1 - 1 \\ 2 | 0 - 1 \end{array}$$

$$(52)_{10} = (110100)_2$$

A binary parallel adder is a digital function that produces the arithmetic sum of two binary numbers in parallel.

The figure below is a BCD to binary code converter implemented with four-bit parallel adders 74HC83. Determine the output for each adder and the final binary output if the decimal 37 is applied to the BCD inputs. Remember that the bits in the BCD representation have decimal weights that are 8, 4, 2, 1 within each code group, but differ by a factor of 10 from one code group to the next, as exemplified in the table next to the figure.



Bit BCD	decimal weight	binary equivalent						
		b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
A ₀	1	0	0	0	0	0	0	1
B ₀	2	0	0	0	0	0	1	0
C ₀	4	0	0	0	0	1	0	0
D ₀	8	0	0	0	1	0	0	0
A ₁	10	0	0	0	1	0	1	0
B ₁	20	0	0	1	0	1	0	0
C ₁	40	0	1	0	1	0	0	0
D ₁	80	1	0	1	0	0	0	0

2.b)

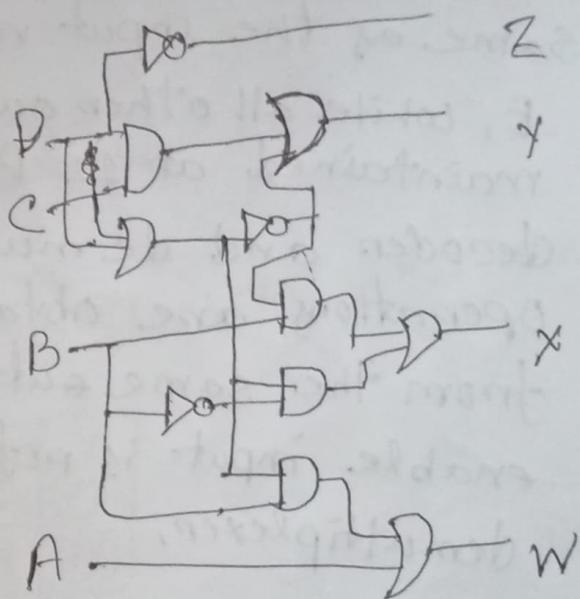
BCD(8421)				Binary Excess-3			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

$$W = A + BC + BD$$

$$X = B'C + B'D + BC'D'$$

$$Y = CD + C'D'$$

$$Z = D'$$



- 3.
- a) Decoder: A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.

Demultiplexer: A demultiplexer is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines.

A decoder with an enable input can function as a Demultiplexer. The selection of a specific output lines is controlled by the bit values of n Selection lines.

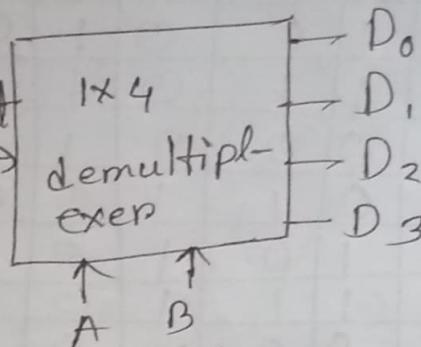
For example .

if the selection lines AB=010,
output D₂ will be the
Same as the input value E

E, while all other outputs are
maintained at 1. Because

decoder and demultiplexer
operations are obtained

from the same output circuit, A decoder with an
enable input is referred to as a decoder /
demultiplexer.



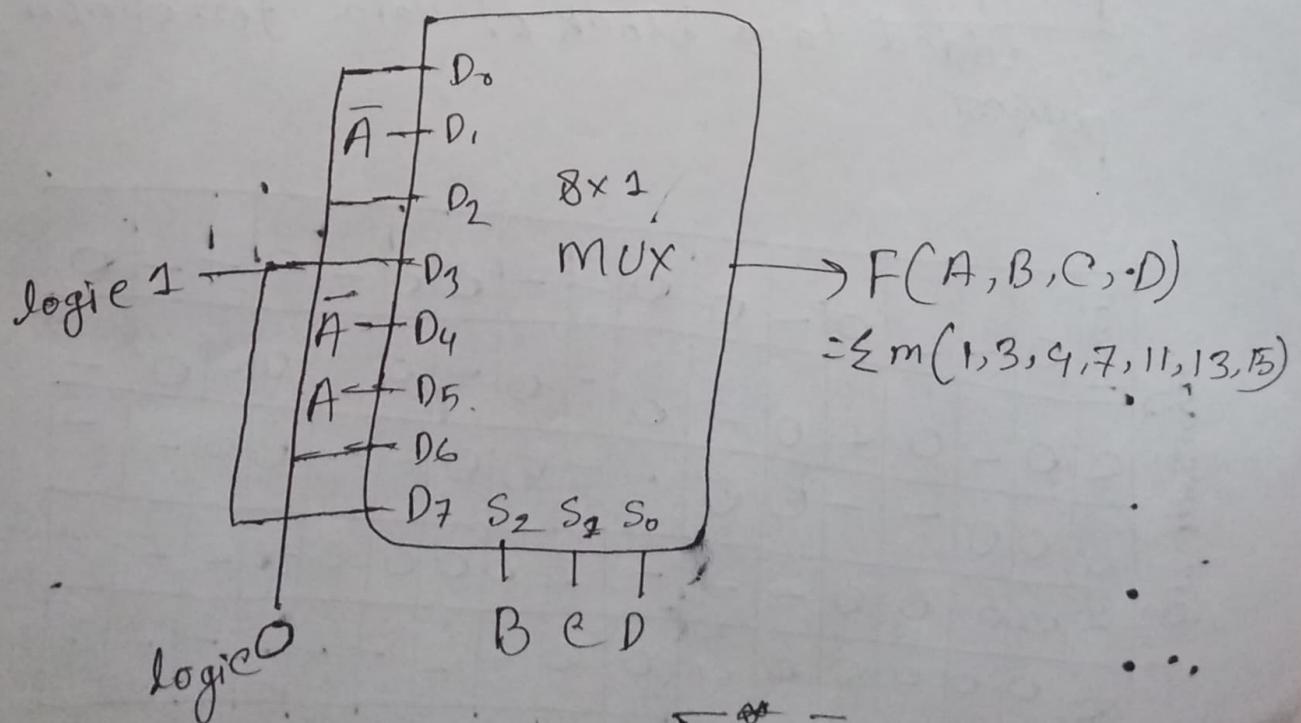
3.b) Multiplexer: A Multiplexer also known as a data selector, is a device that selects between several analog or digital input signals and forwards the selected input to a single output line.

implementation of 8:1 mux

Q

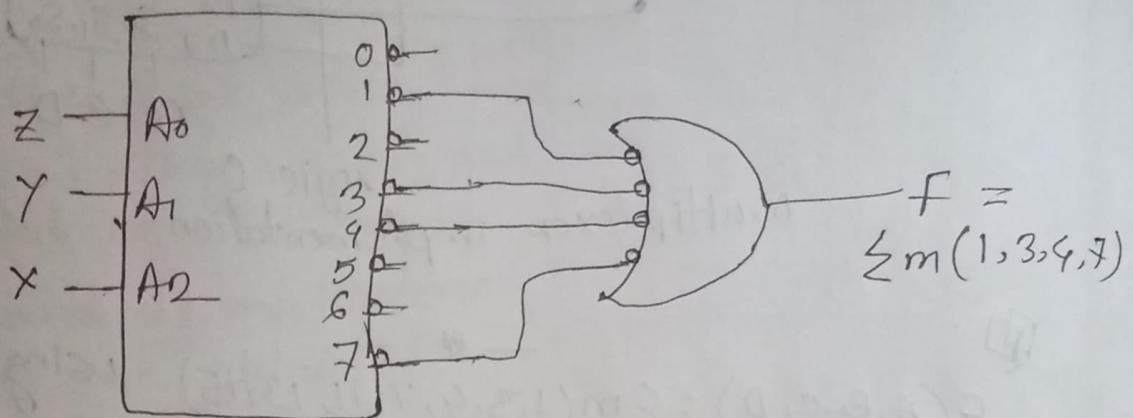
$$F(A, B, C, D) = \sum m(1, 3, 4, 7, 11, 13, 15) \text{ using } 8 \times 1 \text{ MUX}$$

	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
\bar{A}	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	0	\bar{A}	0	1	\bar{A}	A	0	1



3.c) $F(A, B, C) = \sum m(1, 3, 4, 7)$ using a 3-to-8 line decoder with active low output.

A 3-to-8 line decoder, with active low outputs, is used to implement a 3-variable Boolean function f shown in the figure.



4.

a) parity bit: A parity bit is a check bit, which is added to a block of data for error detection purpose.

4-bit message				P _{even}
A	B	C	D	P _{even}
0	0	1	1	0
0	1	0	1	1
0	0	1	0	0
0	0	0	1	1
0	0	0	0	0
0	0	0	0	0

$$P = A \oplus B \oplus C \oplus D$$

2017

1. $200_{10} = 11001000_2$
 a) $(-200)_{10}$ Sign-magnitude =

The least number that is

- 1) a power of 2.
- 2) and is larger than the actual length.
- 3) so that the first bit (leftmost) could be zero (we deal with a positive number at this moment) == by 16.

$$200_{10} = 0000\ 0000\ 1100\ 1000$$

$$= 1111\ 1111\ 0011\ 0111$$

+1

$$-200_{10} = \overline{1111\ 1111\ 0011\ 1000}$$

$$1's \text{ Comp} = 0000\ 0000\ 1100\ 0011$$

$$2's \text{ - } = \overline{0000\ 0000\ 1100\ 1000}$$

2.

a) 200_{10} Sign binary number = 11001000

$$1's \text{ Complement} - - - = 00110111$$

$$2's \text{ - } - - - = \overline{00111000}$$

-

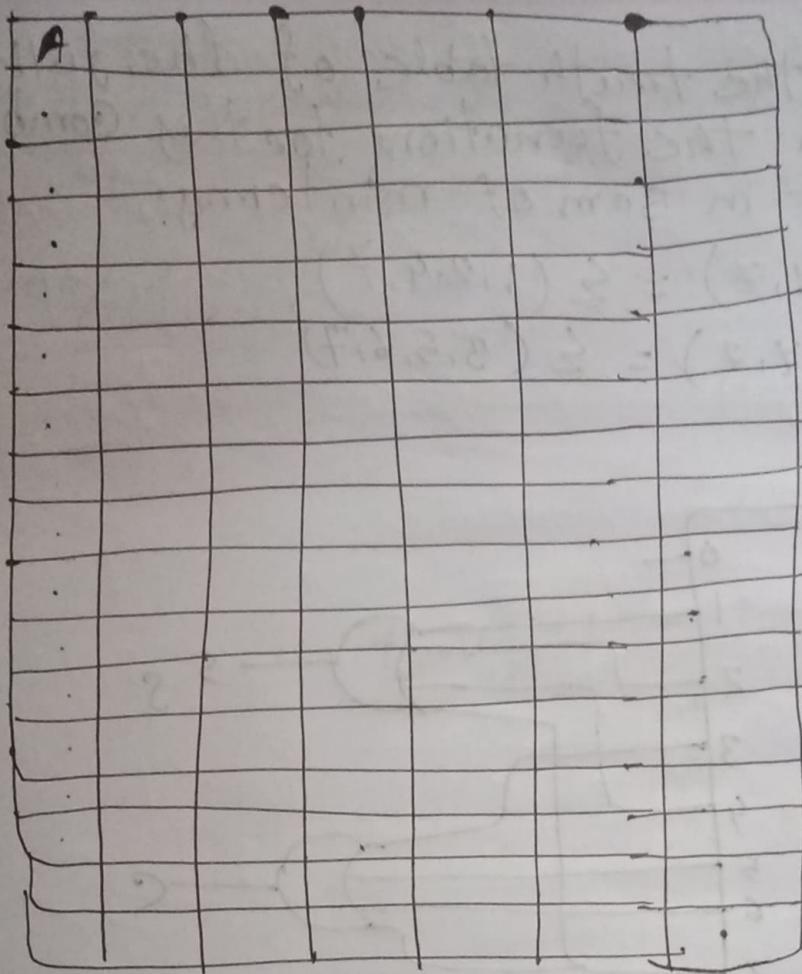
1.
 b)
$$\begin{array}{r}
 & 2 & 7 & 5 & & & 6 & 9 & 1 \\
 & \downarrow & \downarrow & \downarrow & & & \downarrow & \downarrow & \downarrow \\
 0010 & 0111 & 0101 & & 0110 & 0100 & 0001 \\
 0110 & 0100 & 0001 \\
 \hline
 1000 & 1011 & 0110 \\
 8 & 11 & 6 \\
 + 0110 \\
 \hline
 1000 & 10001 & 0110 \\
 11 & 1 & 6 \\
 1000 & 0001 & 0110 \\
 \hline
 1001 & 00010 & 0110
 \end{array}$$

275_{10}
 691_{10}
 $\overline{916}_{10}$

(Ans.)

②)

2.a)



2.b) 2021 20. 4. a

2.c)

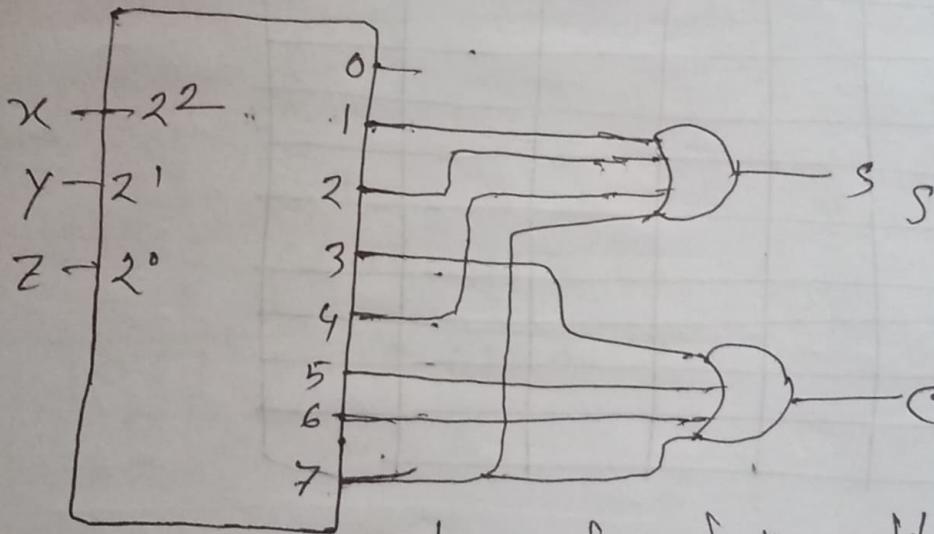
1	0	1	0	0	0
0	0	0	0	0	0
1	0	1	0	0	0
1	0	0	1	0	0
0	1	1	1	0	0
0	1	0	1	0	0
1	0	0	0	1	0
0	1	1	0	1	0
0	1	0	1	1	0
1	1	1	1	1	1

2017

3.b) From the truth table of the full-Adder, we obtain the functions for the combinational circuit in sum of minterms:

$$S(x, y, z) = \Sigma(1, 2, 4, 7)$$

$$C(x, y, z) = \Sigma(3, 5, 6, 7)$$



Implementation of a full-adder with a decoder

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Truth table of full adder

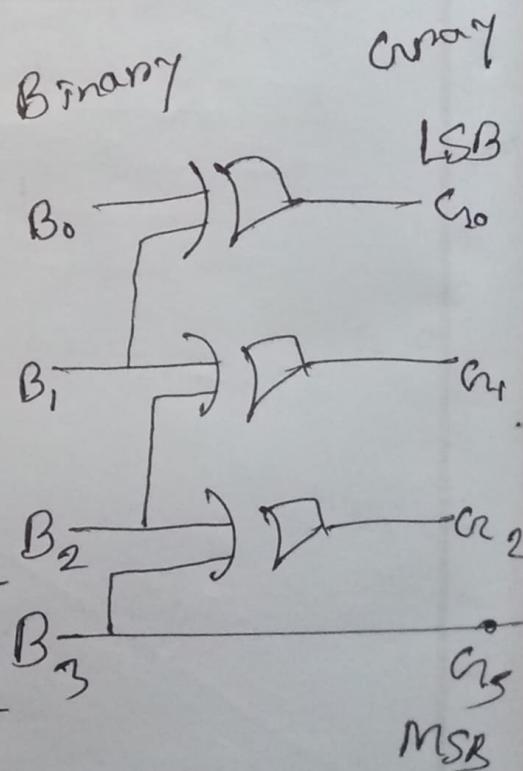
3.a) prop.

Priority encoder: A priority encoder is a circuit or algorithm that compresses multiple binary inputs into a smaller number of outputs.

4.a) Code Converter: A code converter is a logic circuit that changes data presented in one type of binary code to another type of binary code. Such as BCD to Binary, BCD to 7-segment code.

Binary to Gray Conversion;

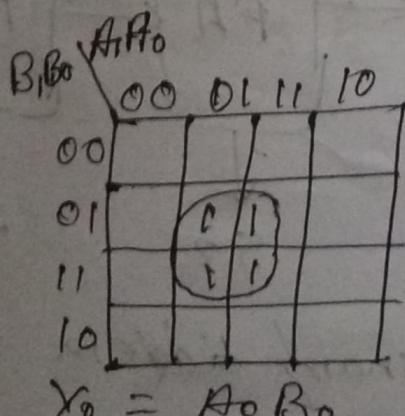
Binary				Gray				
B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀	G ₊
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	
0	1	0	1	0	1	1	0	
0	1	1	0	0	1	1	1	
0	1	1	1	0	1	0	1	
1	0	0	0	1	1	0	0	
1	0	0	1	1	1	0	1	
1	0	1	0	1	1	1	1	
1	0	1	1	1	1	1	0	
1	1	0	0	1	0	1	0	
1	1	0	1	1	0	1	1	
1	1	1	0	1	0	0	1	
1	1	1	1	1	0	0	0	



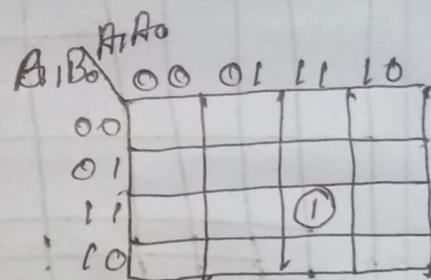
q.b) A multiplier is a combinational logic circuit that we use to multiply binary digits.

First we will draw and using k-map we will derive the boolean expression for all outputs.

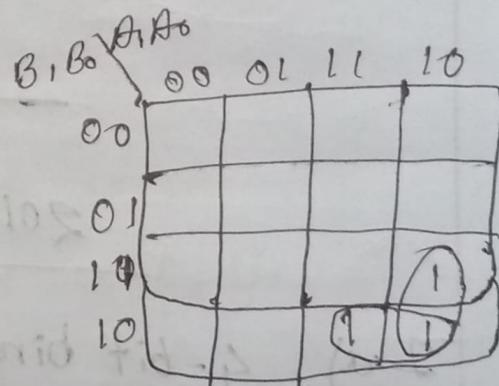
A	B	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	1	0	0	0
0	0	1	0	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	1
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	1	0
1	1	1	1	1	0
1	1	1	1	1	1



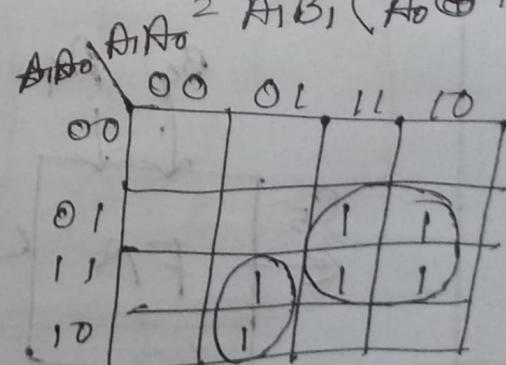
$$Y_0 = A_0 B_0$$



$$Y_3 = A_1 A_0 B_1 B_0$$



$$Y_2 = A_1 \bar{A}_0 B_1 + A_1 B_1 \bar{B}_0$$

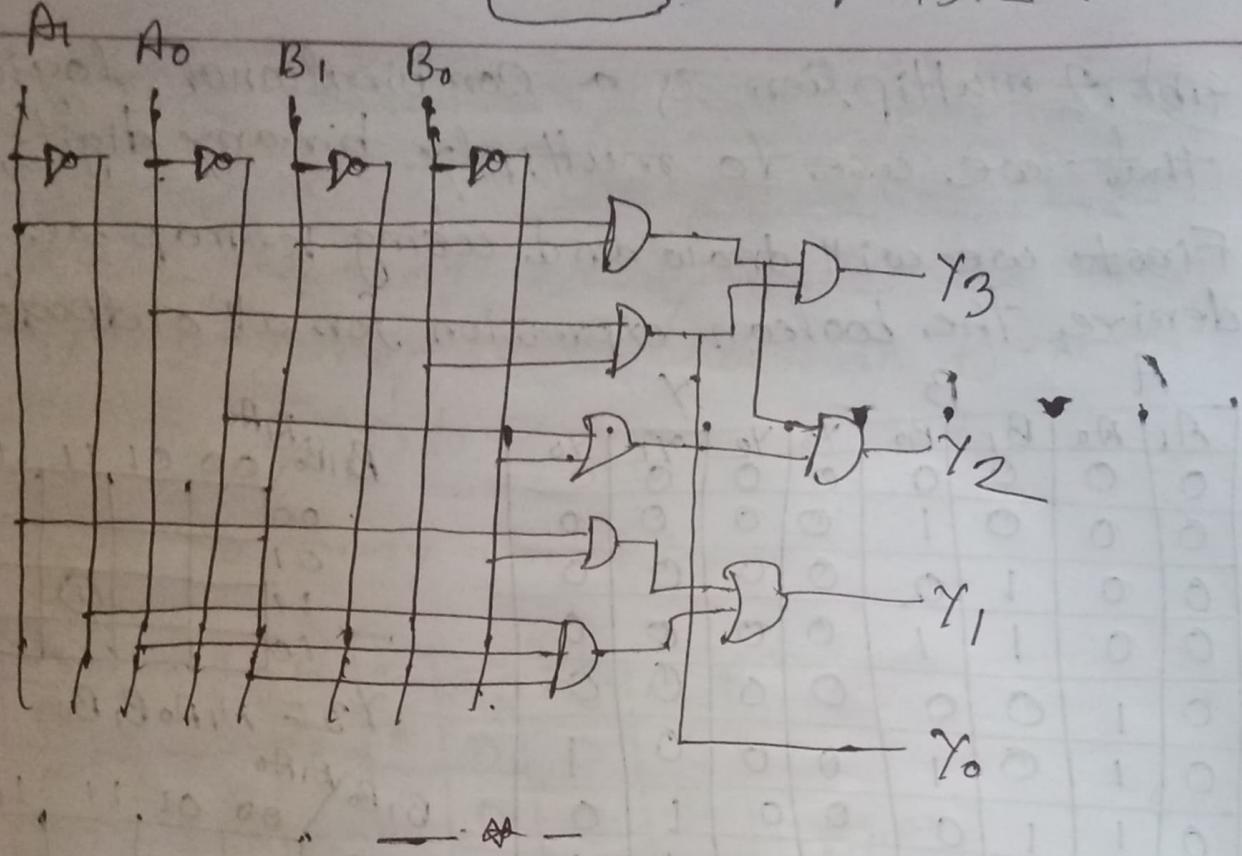


$$Y_1 = A_1 B_0 + \bar{A}_1 A_0 B_1$$

$$A \Rightarrow A_1 A_0$$

$$B \Rightarrow B_1 B_0$$

$$Y = Y_3 Y_2 Y_1 Y_0$$



2016

2(a) 4-bit binary adder circuit

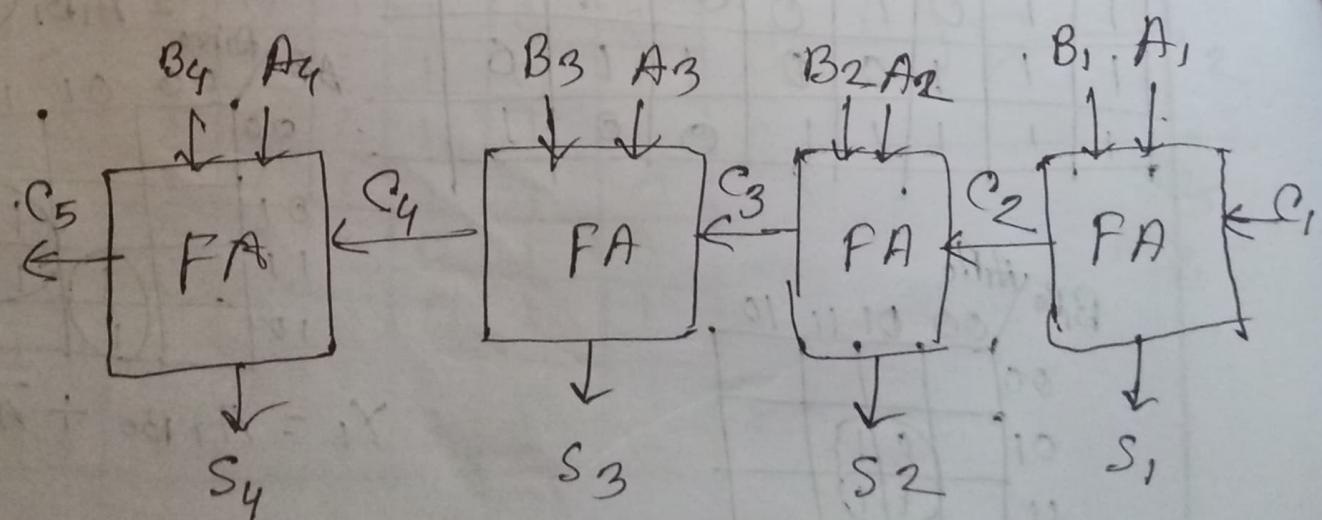
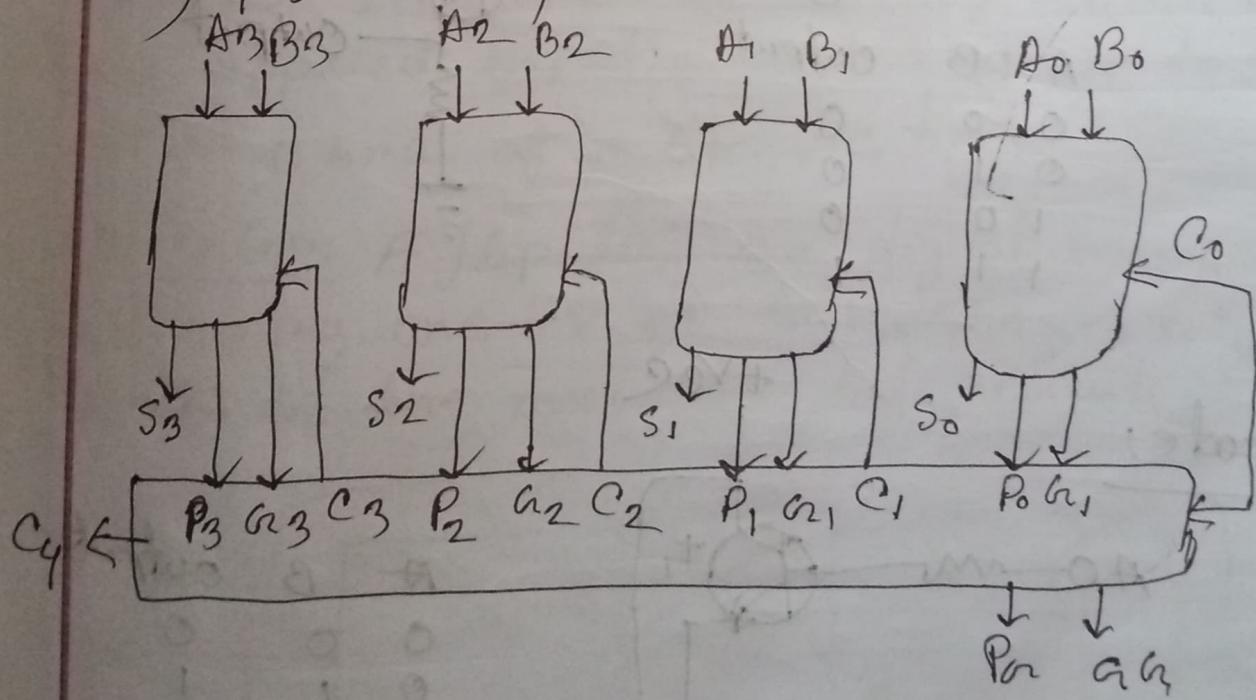


Fig: 4-bit binary adder.

1.b) The carry propagator is propagated to the next level whereas the carry generator is used to generate the output carry, regardless of input carry.

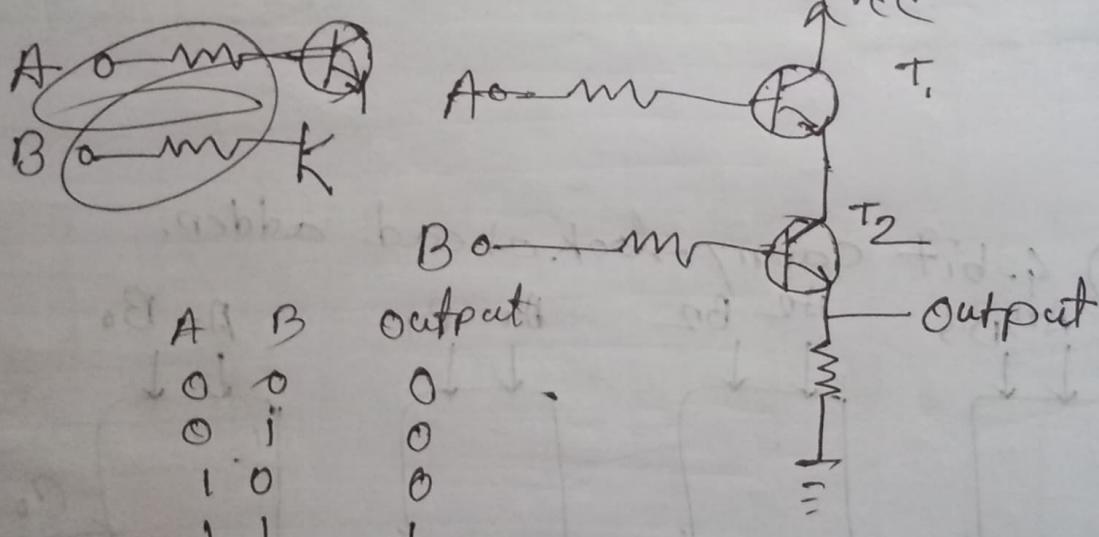
1.c) 4-bit carry-look ahead adder.



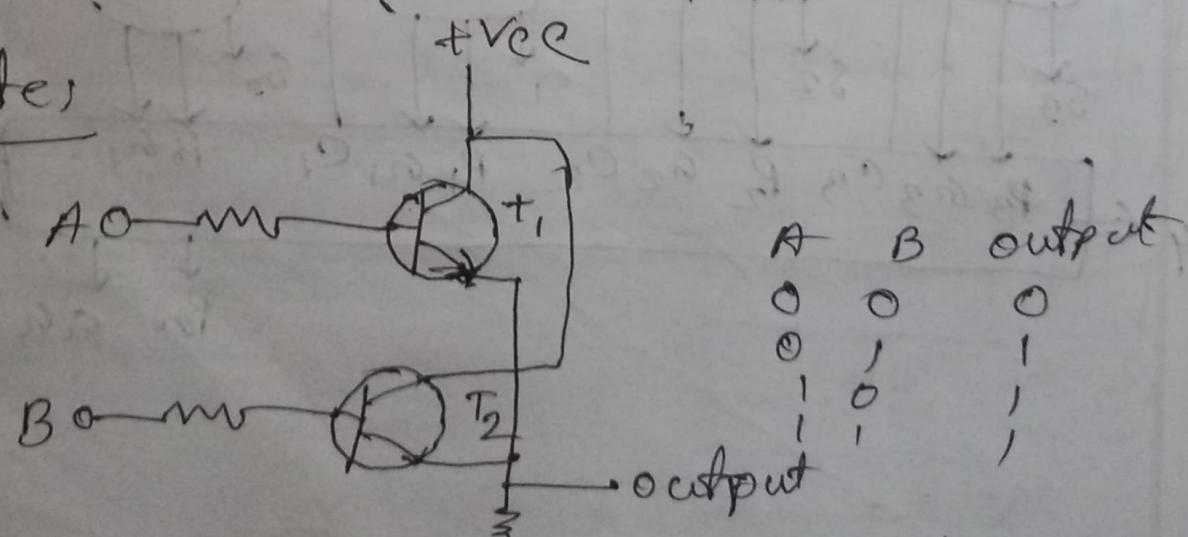
A carry-lookahead adder (CLA) or fast adder is a type of electronic adder used in digital logic. A carry-lookahead adder improves speed by reducing the amount of time required to determine carry bits.

- * -

2. a) 2-input Transistor AND gate



OR gate:



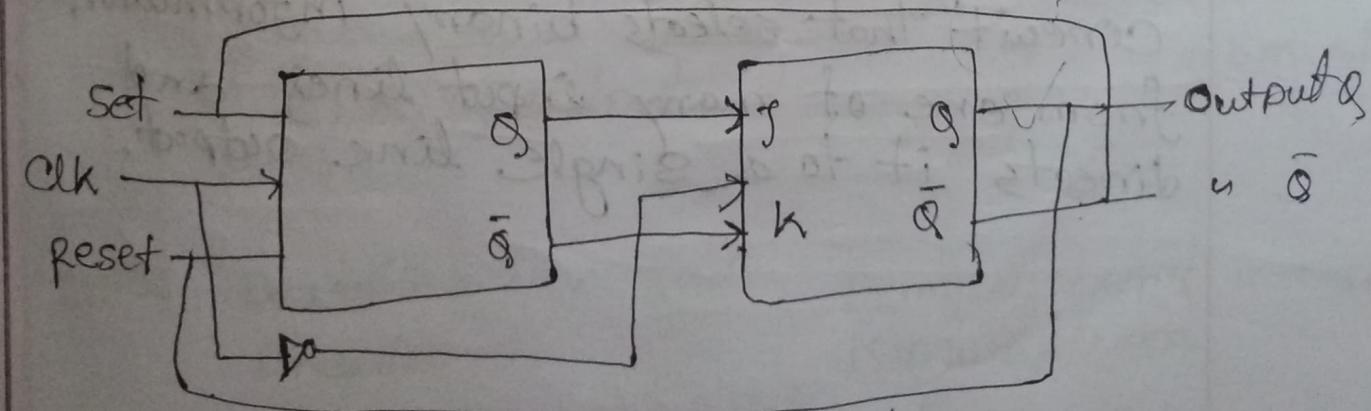
2.c)

Ques. What is meant by signal transmission? Explain with the help of two examples.

3.

b) Latch: A Latch is an electronic device that changes its output immediately on the basis of the applied input. One can use it to store either 0 or 1 at a specific time.

Flip-flop: A flip-flop is a digital memory circuit that stores one bit of data. They are the primary blocks of the most sequential circuits.



2015

1. a) Combinational logic is used in Computer circuits to perform Boolean algebra on input signals and on stored data.

BCD to Excess-9

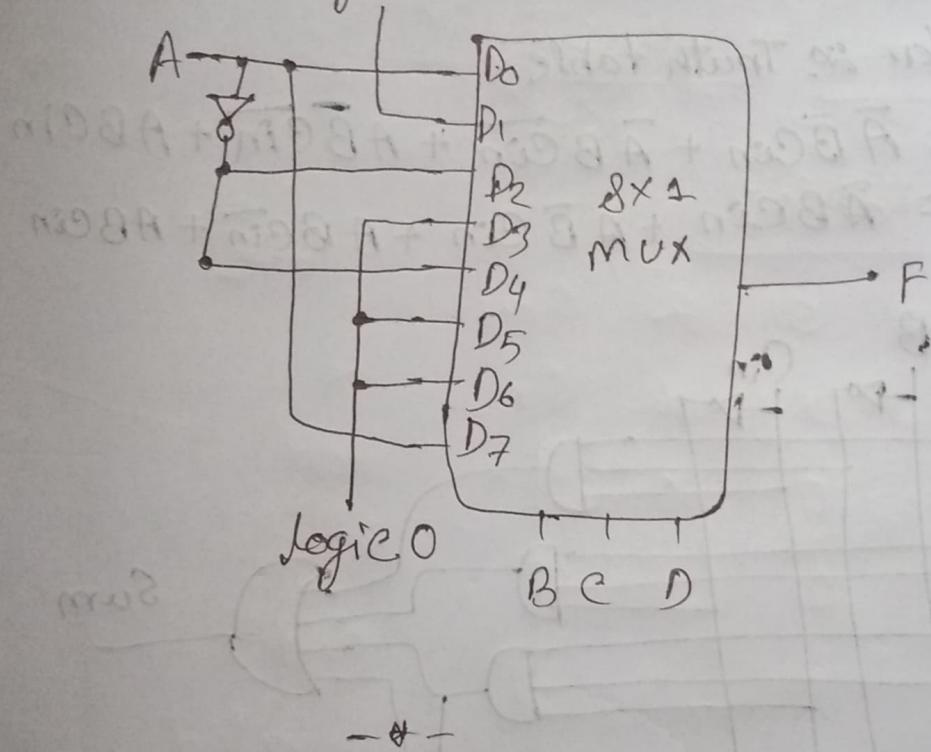
BCD				Excess-9			
B ₃	B ₂	B ₁	B ₀	E ₃	E ₂	E ₁	E ₀
0	0	0	0	0	1	0	0
0	0	0	1	0	1	0	1
0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	1
0	1	0	0	1	0	0	0
0	1	0	1	1	0	0	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

Ans (2015)
Circuit draw
by [unclear]

2. b) A digital multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single line output.

$$F(A, B, C, D) = \Sigma (1, 2, 4, 8, 9, 15) \text{ with a } 8 \times 1 \text{ mux}$$

	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	
A	0	1	2	3	4	5	6	7	
A	8	9	10	11	12	13	14	15	
	A	1	A	0	A	0	0	A	
	logic 1								



2(a)

input/
output

A decoder has input lines and a maximum of $\{2\} \times \{n\}$ output lines.

Inverse Decoder's inverse is Encoder.

De-multiplexer

A Demultiplexer has single input, selection lines and maximum of $\{2\} \times \{n\}$ outputs.

Demultiplexer's inverse is Multiplexer.

Usage - Decoders are used to detect bits, encoding of data

Demultiplexers are used in switching, data distribution,

Select lines

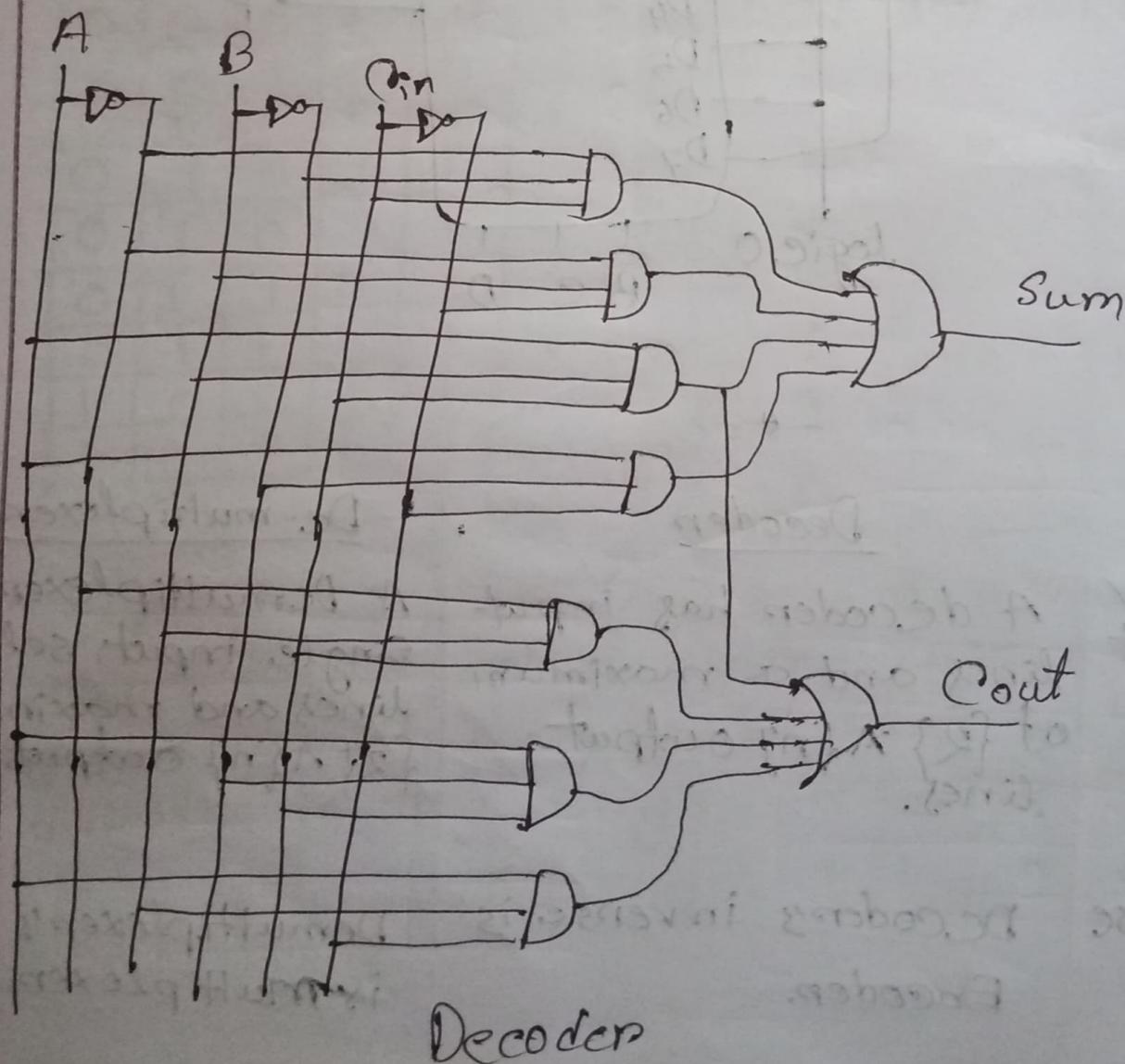
Decoders have no select lines.

Demultiplexers contain select lines.

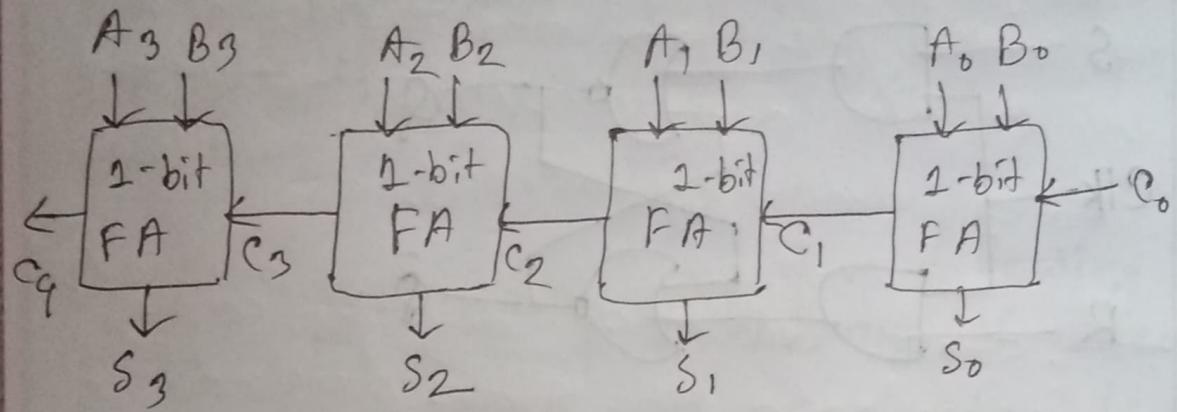
Full adder Truth table

$$\text{Sum} = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

$$\text{Cout} = \bar{A}B\bar{C}_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$



1.
b) reducing the amount of time required to determine carry bits.



4.a)

Flip-Flop

It is a edge triggered device.

Latch

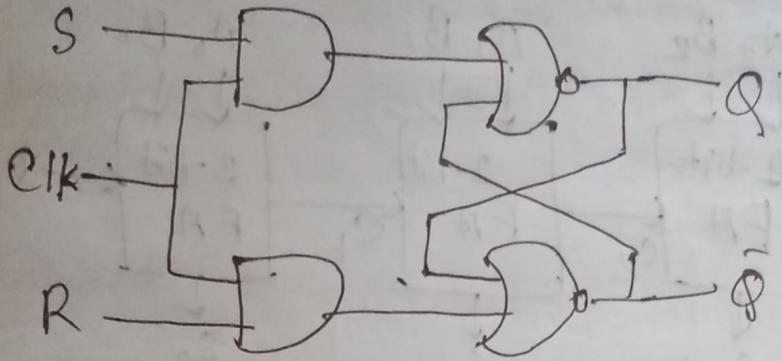
It is a level triggered device.

These are classified into asynchronous or synchronous. Latches don't have a classification in latch.

A flip-flop always have a clock signal.

Latches doesn't have a clock signal.

SR Flip-flop Circuit diagram



Truth table:

Clock	S	R	Q_{n+1}	State
0	x	x	Q_n	x
1	0	0	Q_n	Hold
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	x	Invalid

Characteristic table

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

1	0	0	1
1	0	1	1
1	1	0	x
1	1	1	x

2016

3.c) State reduction problem: The reduction of the number of flip-flops in a sequential circuit is referred to as the state reduction problem.

Don't care condition has the following significance with respect to the digital circuit design: Simplification: These conditions denote the set of inputs that never occurs for given digital circuits. Therefore, to simplify the boolean output expression, the "don't care" are used.



2018

4.b) The state diagram is a graphical representation of a sequential circuit. The state table contains the same information as the state diagram in tabular form.

2018

q1b) The state diagram is a graphical representation of a sequential circuit in which the states are represented by circles and transitions between states shown by arrows.

The state table contains the same information as the state diagram in tabular form.

Left \rightarrow MSB \leftarrow Right \rightarrow MSB
2020

1.

a) We use binary number system instead of decimal number system in the digital devices are Simplicity, Consistency, Compatibility and Efficiency etc.

b)

i) Sign magnitude: The representation of decimal numbers in everyday business is commonly called the signed-magnitude representation.

93 \rightarrow 2's Complement

93 = 01011101 is Complement

$$\begin{array}{r} \text{Signbit} \quad \swarrow \\ 10100010 \quad \curvearrowright \\ +1 \\ \hline 1010011 \end{array}$$

2's Complement

Sign magnitude =

$$-93 = 10100010$$

Complement, 01011101

b) 1's Complement

2. To get 1's complement of a binary number, simply invert the given number.

2. 1's complement of binary number —
110010 is 001101.

3. Simple implementation which uses only NOT gates for each input bit.

4. Can be used for signed binary number representation but not suitable as ambiguous representation for number 0.

5. 1's Complement arithmetic operations are not easier than 2's Complement because of addition of end-around-carry bit.

6. To get 2's complement of a binary number simply invert the given number and add 1 to the least significant bit (LSB) of given result.

2. 2's complement of binary number —
110010 is 001110.

7. Uses NOT gate along with full adder for each input bit.

8. Can be used for signed binary number representation and most suitable as unambiguous representation for all numbers.

9. 2's complement arithmetic operations are much easier than 1's complement because of these if no addition of end-around-carry-bit.

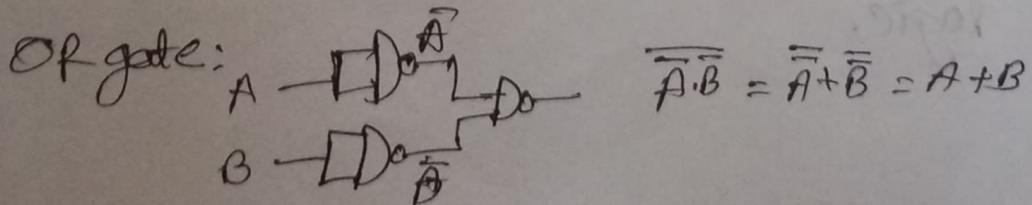
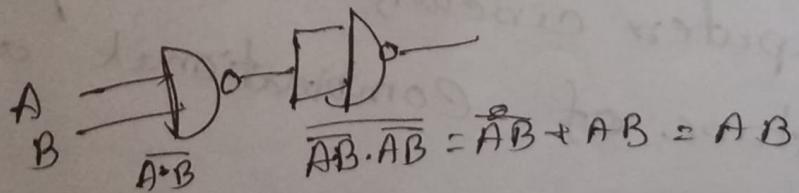
2020

2.a) A universal gate is a logic gate which can implement any Boolean function without the need to use any other type of logic gate. The NOR gate and NAND gate are universal gates. This means that you can create any logical Boolean expression using only NOR gates or only NAND gates.

NOR gate is a not gate.

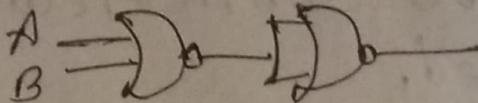
$A \rightarrow D \rightarrow \overline{A}$ is a not gate.

Now we will see the design of an AND gate from NAND gate.



Again, nor gate:

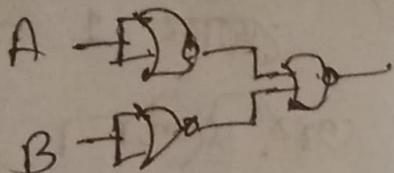
OR



$$\overline{A+A}$$

$$\overline{\overline{A+B} + \overline{A+B}} = (\overline{A+B}) \cdot (\overline{A+B}) = \overline{A+B}$$

AND,



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

NOT gate



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

— A —

to BCD to Decimal to Binary