Unit: 3

Combinational logic circuit.

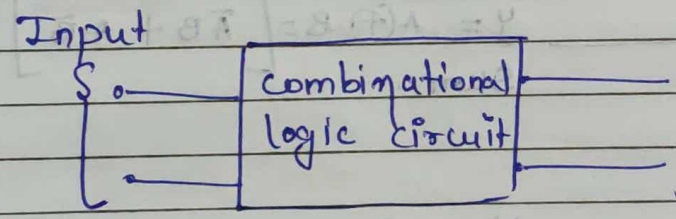
Important Topic

1. What is Combinational Logic circuit.
2. Comparator.
3. Encoder & Decoder.
4. Multiplexer & De-multiplexer.
5. Adder & full Adder.
6. Half Subtractor & full Subtractor.
7. parity Generator.

Q. What is Combinational Logic Circuit?

↳ Combinational Logic circuit is the Device made from logic gates which only depends on the present input.

↳ The Output is not stored anywhere. As There is NO MEMORY.



↳ It do not depend upon the past output.

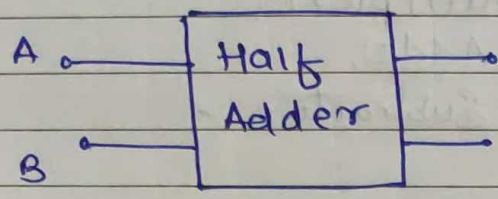
↳ Types of Combination Logic circuit are :-

- Half Adder
- Full Adder
- Half Subtractor
- Full Subtractor
- Encoder / Decoder
- Multiplexer / Demultiplexer
- Comparator.

1. Half Adder :- 1 Bit Addition

→ Input :- 2 (A and B)
→ Output :- 2 (sum and carry)
→ Gates :- Ex-OR, AND gate.

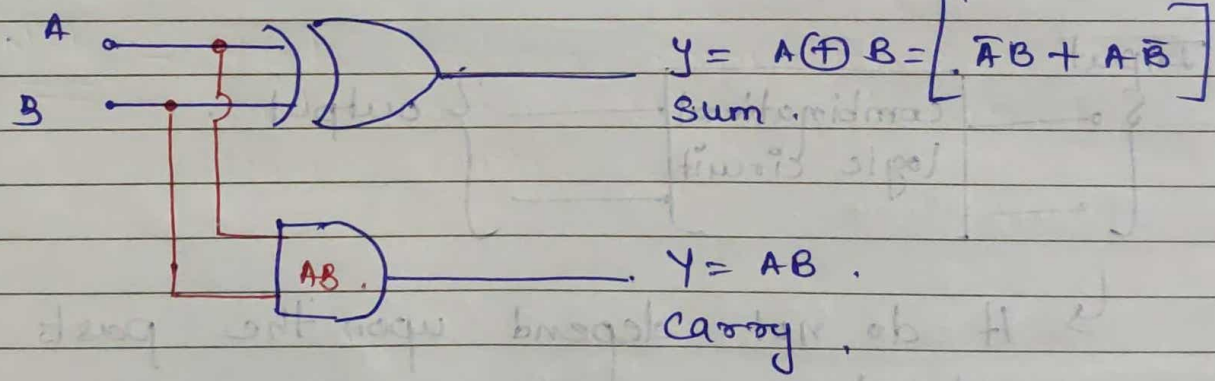
(a). Block Diagram.



(b) Truth Table :-

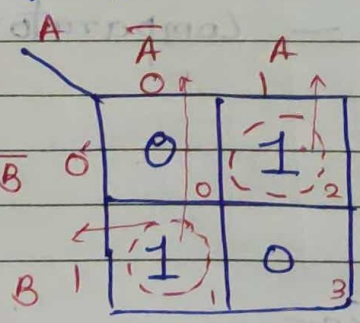
		Sum	Carry
A	B	$Y = A \oplus B$	$Y = AB$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(c) Logic gate Diagram.



(D) K'map for Both Output to get Boolean Expression.

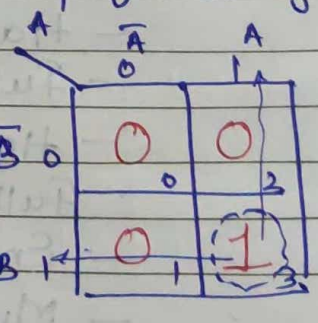
K'map for sum.



Sum :-
Ans :- $\bar{A}B + A\bar{B}$

Carry :- AB

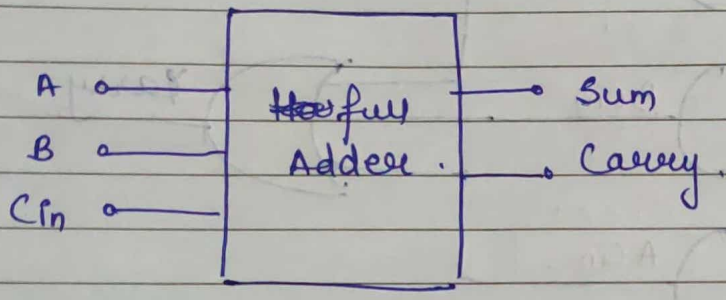
K'map for carry.



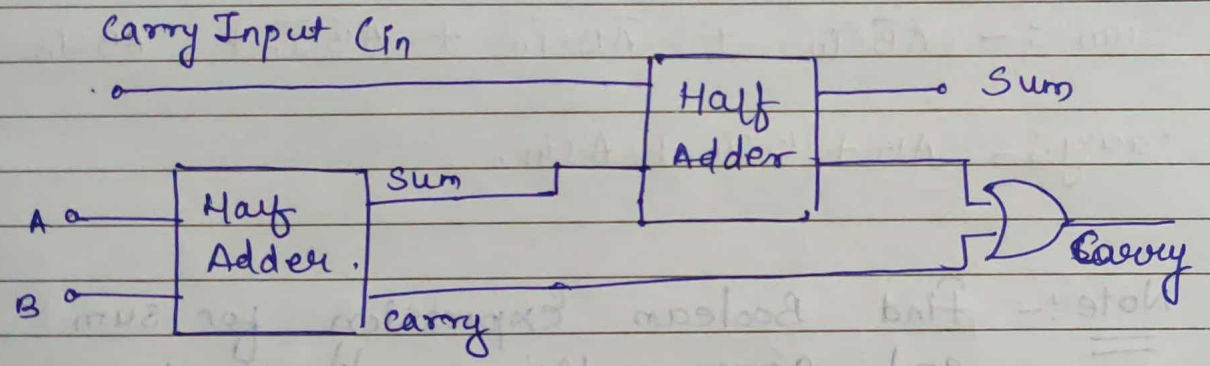
Full Adder :- 2 bit Addition.

Input :- 3 (A, B and C_{in})
Output :- 2 (Sum and Carry).
Total gates :- EX-OR - (1)
AND - (3)
OR - (1).

(a) Block Diagram :-



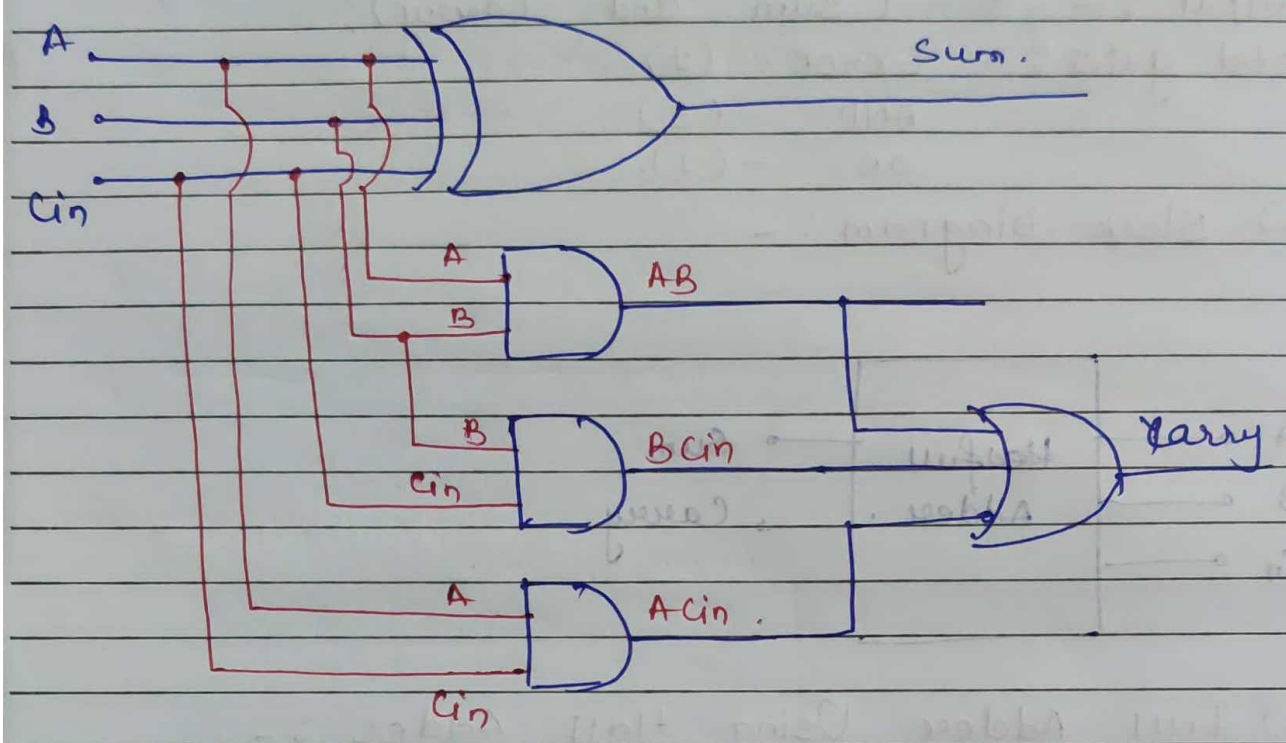
(b) Full Adder Using Half Adder :-



(C) Truth Table.

	A	B	C_{in}	Sum	Carry
$\bar{A}\bar{B}\bar{C}_{in}$	0	0	0	0-0	0
$\bar{A}\bar{B}C_{in}$	0	0	1	0-1	0
$\bar{A}B\bar{C}_{in}$	0	1	0	1-2	0
$\bar{A}BC_{in}$	0	1	1	0-3	1
$A\bar{B}\bar{C}_{in}$	1	0	0	1-4	0
$A\bar{B}C_{in}$	1	0	1	0-5	1
$AB\bar{C}_{in}$	1	1	0	0-6	1
ABC_{in}	1	1	1	1-7	1

(d) Logic Diagram of full Adder :-



Sum :- $\bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

Carry :- $AB + BC_{in} + AC_{in}$.

Note :- Find Boolean Expression for sum and Carry Using K'map. by your own.

ENCODER.

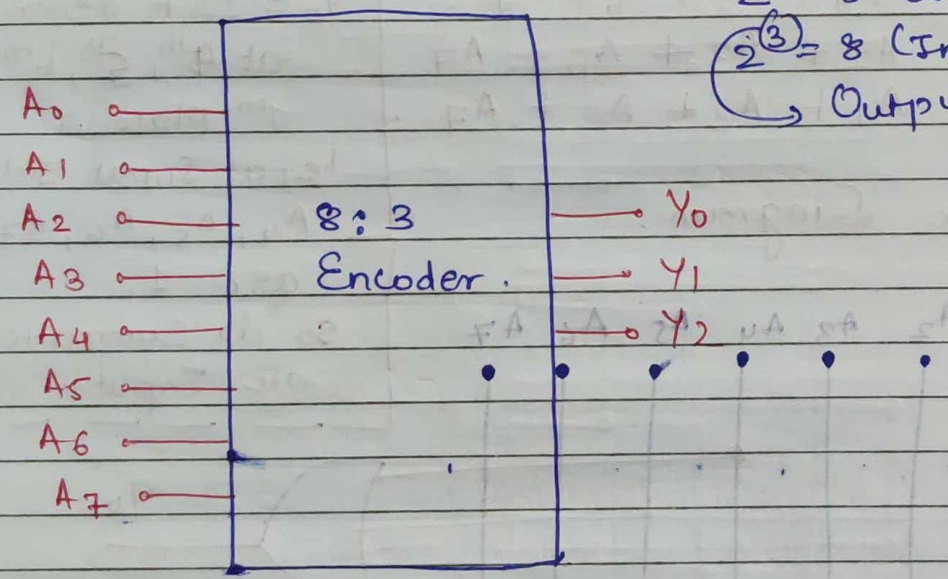
→ It converts Decimal to Binary.
→ Input = 2^n Input lines.
Output = n Output lines

→ 2:1 encoder
→ 4:2 Encoder
8:3 Encoder
16:4 Encoder.

Example :- 8:3 Encoder.
↙ Input ↘ Output.

(a) Block Diagram :-

8 = Input
3 = Output.
?
2^3 = 8 (Input)
2^3 = 8 (Input.
→ Output.



→ Use 2^n formula.

→ where 2^n = Input lines.

n = Output lines.

Try 4:2 and 16:4 encoders by your own.

(b) Truth Table of 8:3 Encoder.

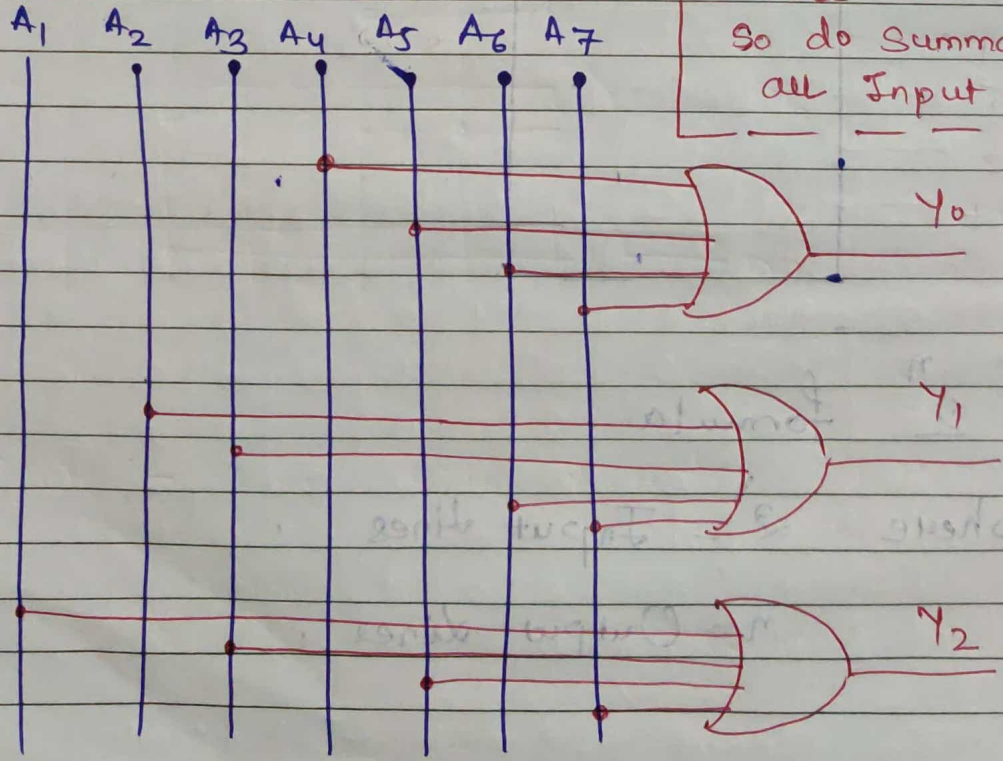
	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Y_0	Y_1	Y_2	
0 =	1	0	0	0	0	0	0	0	0	0	0	=0
1 =	0	1	0	0	0	0	0	0	0	0	1	=1
2 =	0	0	1	0	0	0	0	0	0	1	0	=2
3 =	0	0	0	1	0	0	0	0	0	1	1	=3
4 =	0	0	0	0	1	0	0	0	1	0	0	=4
5 =	0	0	0	0	0	1	0	0	1	0	1	=5
6 =	0	0	0	0	0	0	1	0	1	1	0	=6
7 =	0	0	0	0	0	0	0	1	1	1	1	=7

Note :-
Count No. of 1 in Y_0, Y_1 & Y_2.
Y_0 = 1's is generated at 4th, 5th, 6th & 7th place.
i.e at Input side A_4, A_5, A_6, A_7 are 1.
So do Summation of all Input.

(c) Boolean Expression.

$Y_0 = A_4 + A_5 + A_6 + A_7$.
$Y_1 = A_2 + A_3 + A_6 + A_7$
$Y_2 = A_1 + A_3 + A_5 + A_7$.

(D) Logic Diagram.

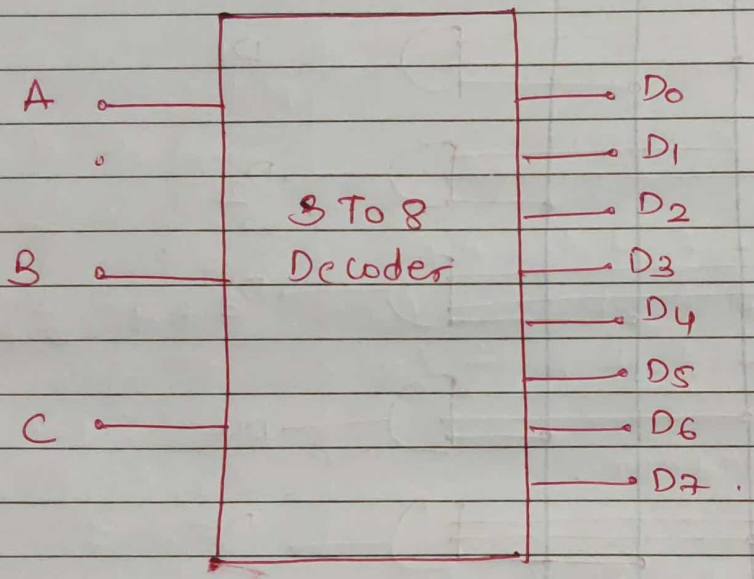


Decoder.

→ Opposite of Encoder.
→ Input = n
Output = 2^n.

→ 1:2 Decoder — 1 = Input & 2 = Output.
→ 2:4 Decoder — 2 = Input & 4 = Output
3:8 Decoder — 3 = Input & 8 = Output.
4:16 Decoder. — 4 = Input & 16 = Output.

(a) Block Diagr 3 To 8 Decoder.



(b) Truth Table :-

A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

The Input and output of Decoder is opposite of encoder truth table.

(C) Boolean Expression.

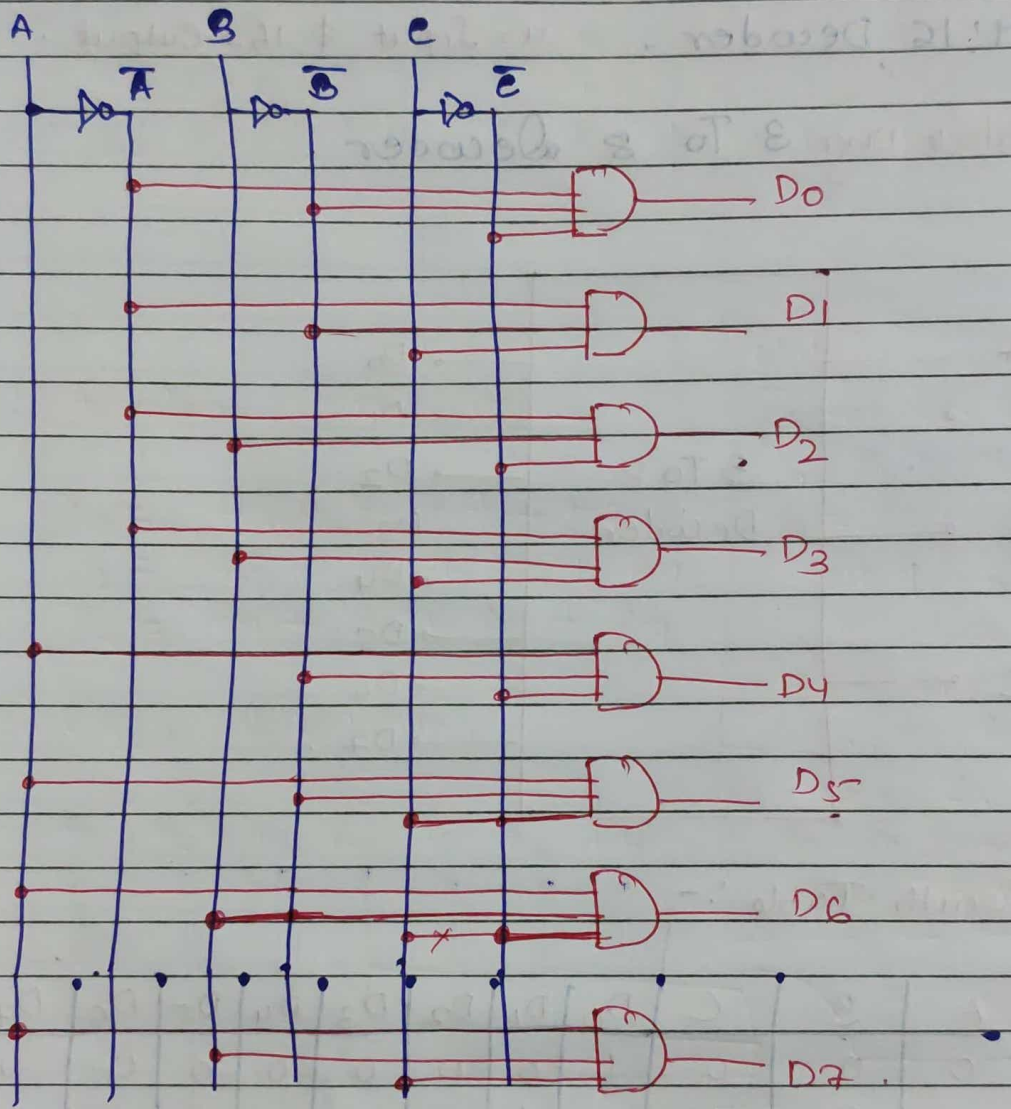
$D_0 = \bar{A}\bar{B}\bar{C}$ $\quad D_4 = A\bar{B}\bar{C}$

$D_1 = \bar{A}\bar{B}C$ $\quad D_5 = A\bar{B}C$

$D_2 = \bar{A}B\bar{C}$ $\quad D_6 = AB\bar{C}$

$D_3 = \bar{A}BC$ $\quad D_7 = ABC$

(D) logic Diagram :-



Note :- Try 4:16 Decoder by your Own.

Multiplexer :- Many Input & One Output.

→ It can take number of Input and gives a single output.

→ Select lines are used which will select which input line will be using multiplexer.

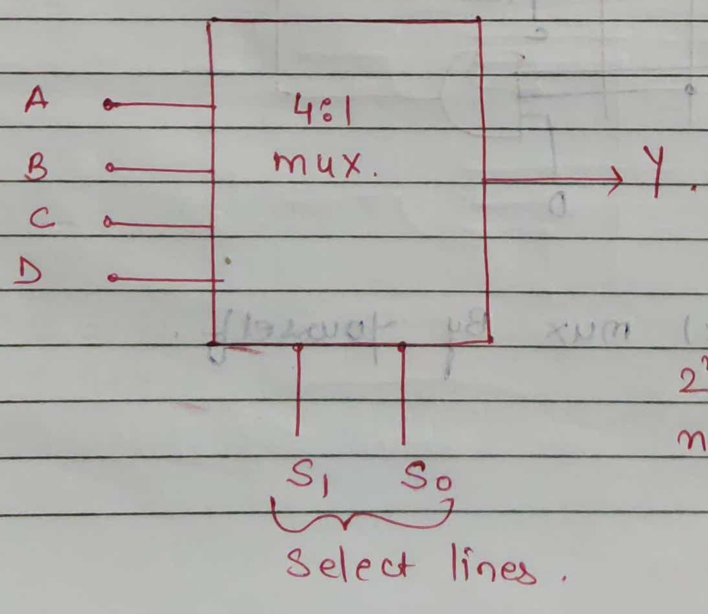
→ 2:1, 4:1, 8:1, 16:1 } multiplexer, where, 2, 4, 8 and 16 = Input.

→ According to Input, Select lines are taken.

Eg:- 4:1 multiplexer,
4 = Input.
1 = Output.
Select lines = ?

Then $2^{?} = 4$.
$2^{2} = 4$. → (2) = Select lines.

(a) 4:1 Multiplexer.



2^n = Input.
n = Select lines.

(b) Truth Table of 4:1 Multiplexer

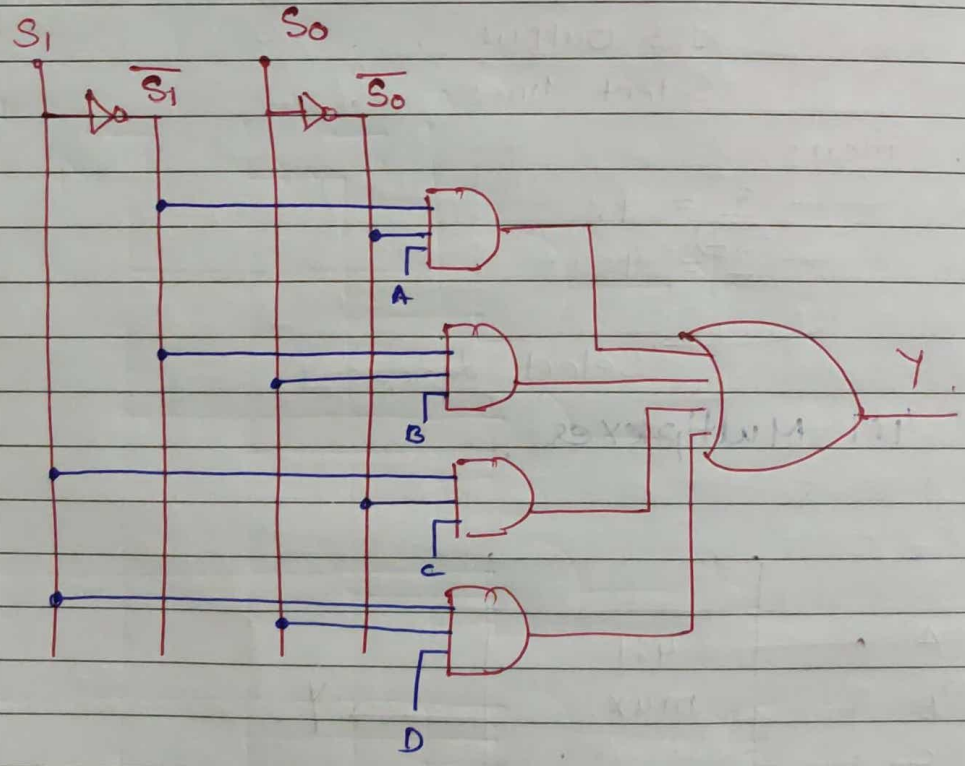
	S_1	S_0	Y
0	0	0	A
1	0	1	B
2	1	0	C
3	1	1	D

(c) Boolean Expression

$$Y = A.\bar{S}_1\bar{S}_0 + \bar{S}_1 S_0 B + C.S_1\bar{S}_0 + S_1 S_0.D$$

$$= \bar{S}_1\bar{S}_0.A + \bar{S}_1 S_0.B + S_1\bar{S}_0.C + S_1 S_0.D$$

(D) Logic Diagram.



Note:- Try 8:1 mux By yourself.

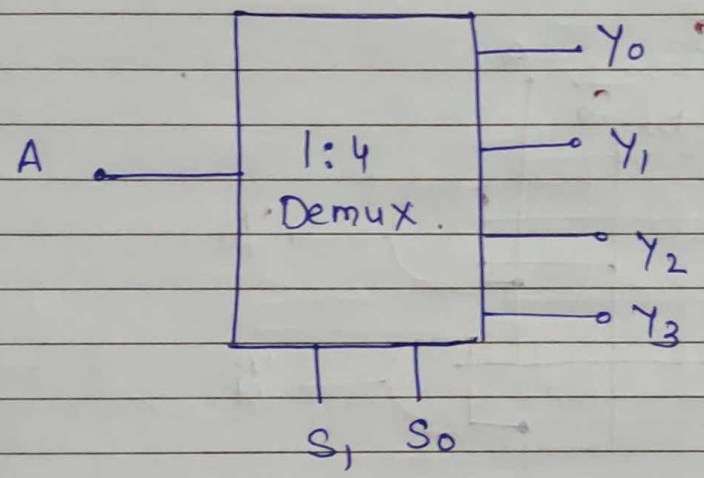
- Opposite of Multiplexer

Demultiplexer.

↳ 1 input and many output.
↳ Input = 1.
~~Iop~~.
Output = 2^n
Select lines = n.

→ 1:2 } Demultiplexer.
1:4
1:8
1:16
→ Select lines will select the output
→ Example (1:4)

(a) 1:4 Demultiplexer :- Block diagram



?
2 = 4 (output)
(2)
2 = 4 (output)
↳ Select lines.

(b) Truth Table of 1:4 Demux.

S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	A	0	0	0
0	1	0	A	0	0
1	0	0	0	A	0
1	1	0	0	0	A

(c) Boolean Expression :-

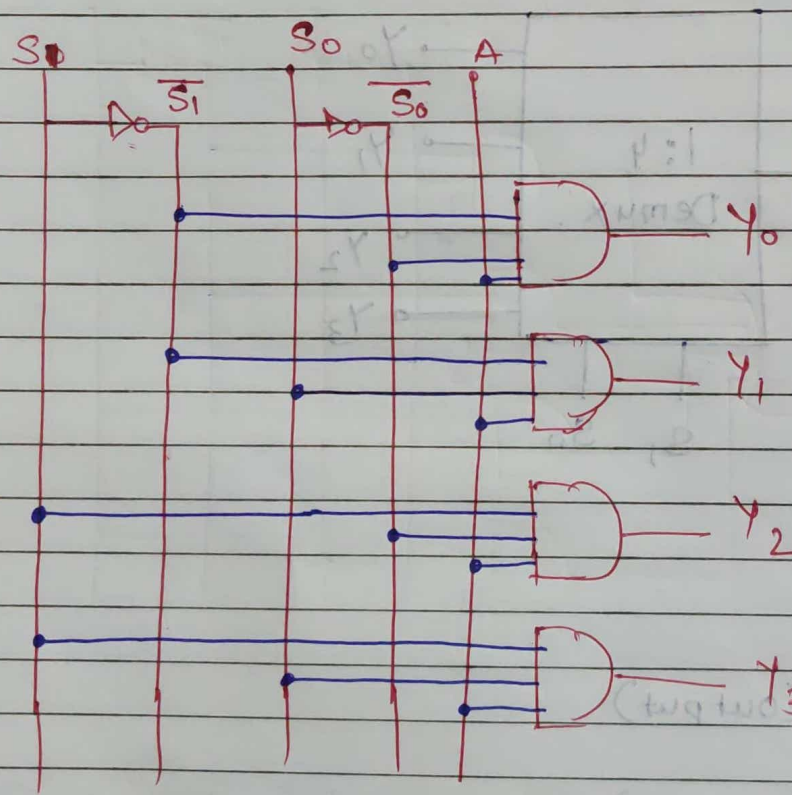
$Y_0 = \bar{S}_0 \bar{S}_1 . A = \bar{S}_1 \bar{S}_0 A$

$Y_1 = \bar{S}_0 S_1 . A = \bar{S}_1 S_0 A$

$Y_2 = S_0 \bar{S}_1 . A = S_1 \bar{S}_0 A$

$Y_3 = S_0 S_1 . A = S_1 S_0 A$

(D) logic Diagram



Note :- Try 1:8 Demux By your Own.

Comparator

↳ It compares the two input either it is greater, smaller or equal.

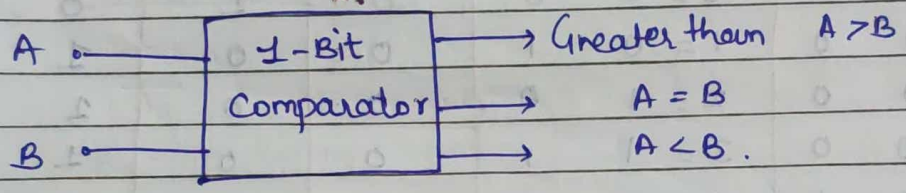
↳ Used in CPU and micro-Controller.

↳ Also called as Magnitude Comparator.

↳ 1 Bit, 2 Bit and 4 Bit magnitude Comp^{r}

↳ X-NOR is a basic Comparator

(a) 1-Bit magnitude. Comparator.



(b) Truth Table.

A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

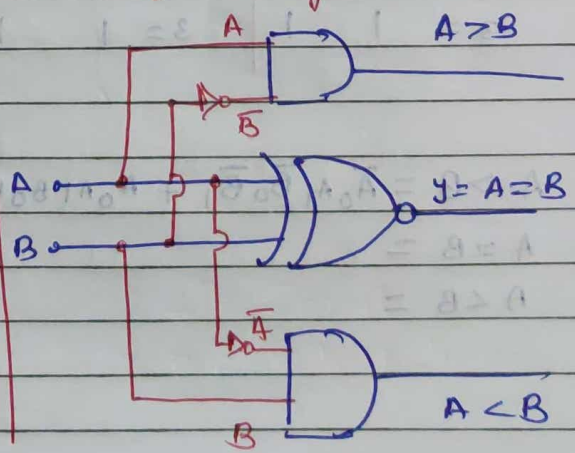
(c) Boolean Expression.

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

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

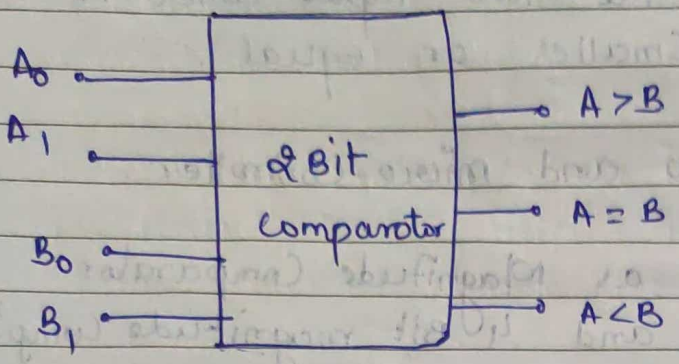
$A<B = \bar{A}B$

(d) Logic Diagram



2 Bit Magnitude Comparator

4 Input
3 output.



Truth Table

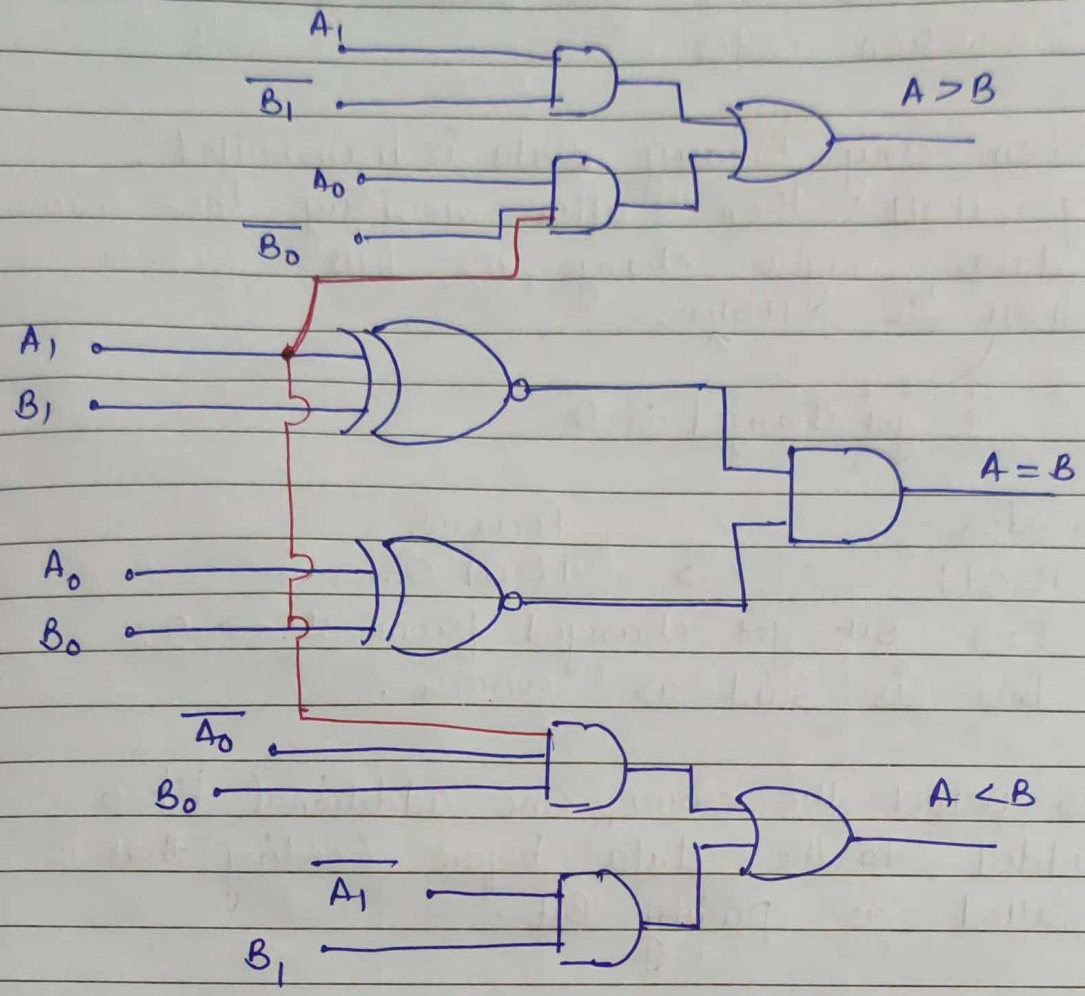
A_0	A_1	B_0	B_1	A>B	A=B	A<B	
0 = 0	0	0 = 0	0	0	1	0	0
0 = 0	0	1 = 0	1	0	0	1	1
0 = 0	0	2 = 1	0	0	0	1	2
0 = 0	0	3 = 1	1	0	0	1	3
1 = 0	1	0 = 0	0	1	0	0	4
1 = 0	1	1 = 0	1	0	1	0	5
1 = 0	1	2 = 1	0	0	0	1	6
1 = 0	1	3 = 1	1	0	0	1	7
2 = 1	0	0 = 0	0	1	0	0	8
2 = 1	0	1 = 0	1	1	0	0	9
2 = 1	0	2 = 1	0	0	1	0	10
2 = 1	0	3 = 1	1	0	0	1	11
3 = 1	1	0 = 0	0	1	0	0	12
3 = 1	1	1 = 0	1	1	0	0	13
3 = 1	1	2 = 1	0	1	0	0	14
3 = 1	1	3 = 1	1	0	1	0	15

$A>B = \bar{A}_0 A_1 \bar{B}_0 \bar{B}_1 + A_0 A_1 B_0 B_1 + A_0 A_1 B_0 B_1 + A_0 A_1 B_0 B_1 + A_0 A_1 B_0 B$

$A=B =$ $+ A_0 A_1 B_0 B_1$

$A<B =$

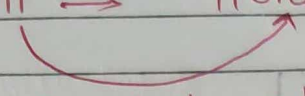
Logic diagram of 2-bit magnitude Comparator.



Parity Generator.

→ Ex-OR is used for error-Detection and Correction codes.

→ When any Binary Data is transmitted, Eg:- 11011 then after receiving the same data might change the bits from 11011 to 11010.



1 got changed to 0.

Sender = 10011 ⟶ Receives 10010.

Last Bit got changed from 1 → 0.
This is said as error.

= To Detect the error, One additional bit is added to the data before Sending that is called as parity Bit.

= Parity Bit checks Single bit error.

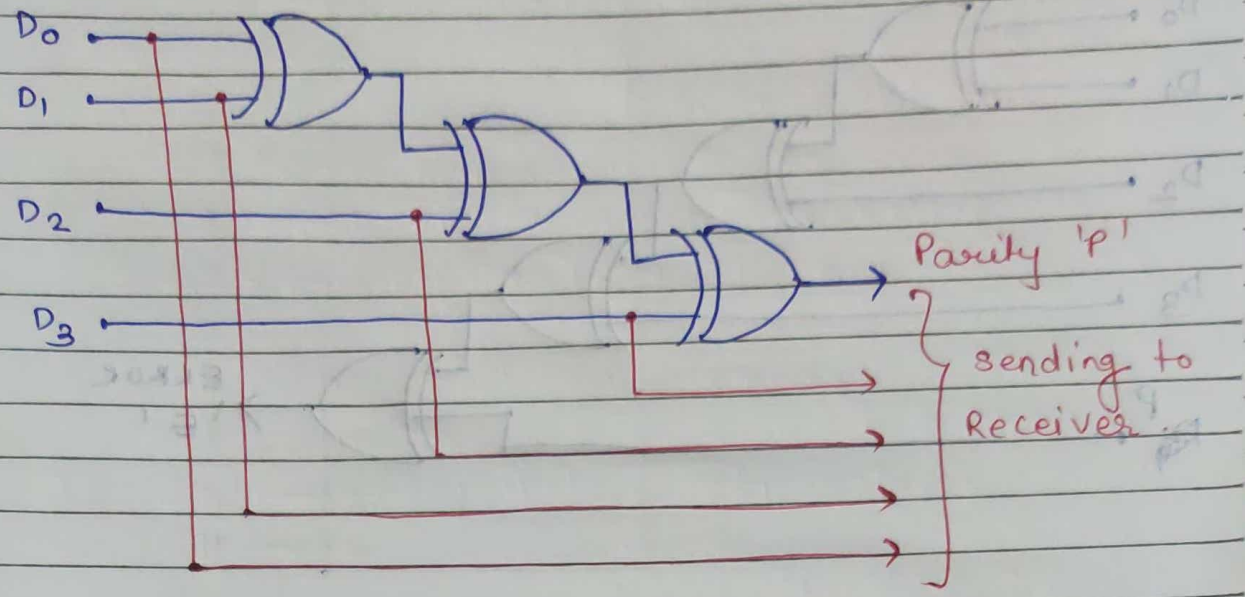
= Parity Generator = It generates parity Bit at transmitted.

= parity checker = It checks the parity in the receiver.

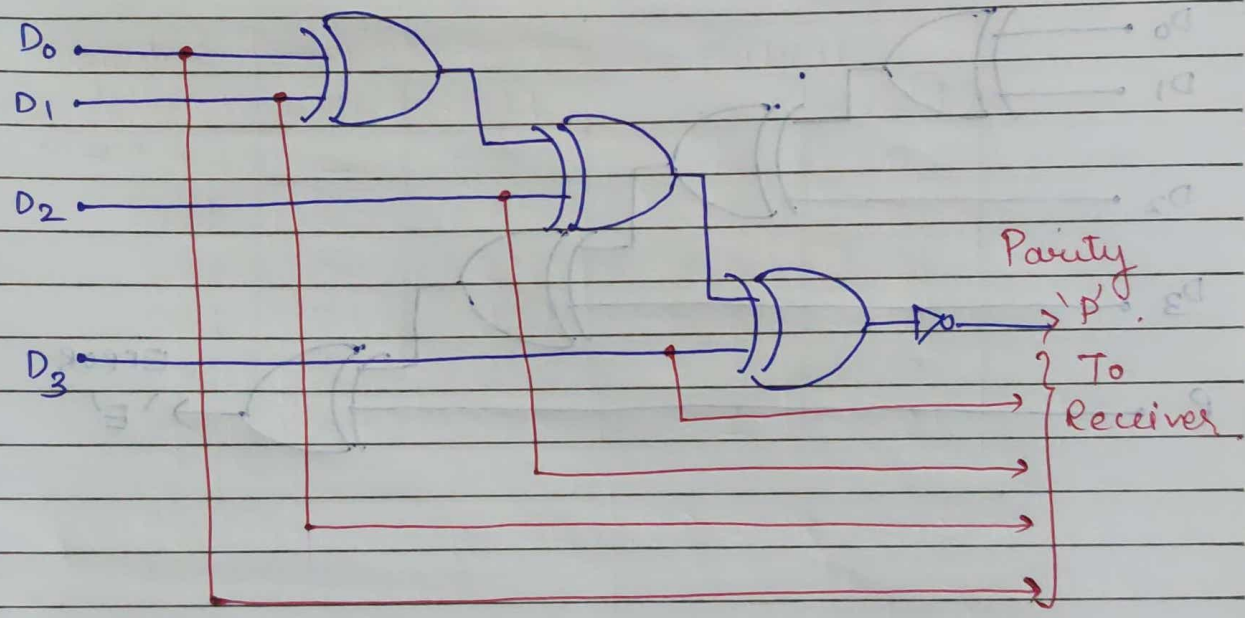
= Even parity = No. of 1's is even in data
= Odd parity = No. of 1's is odd in data.

If even data is received at odd parity = Error.
If in even parity checker, the Received data is having odd parity then there is error.

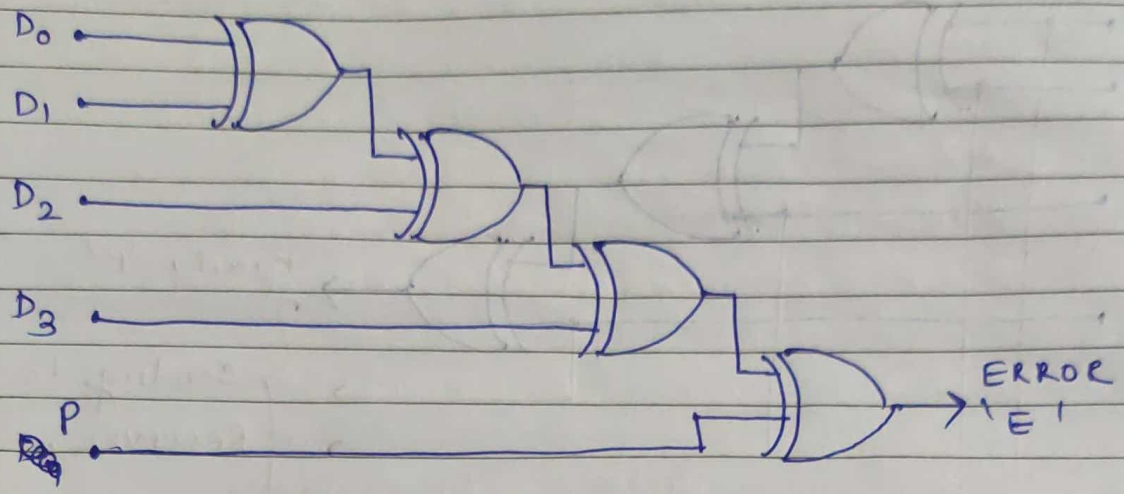
(a) Even parity generator



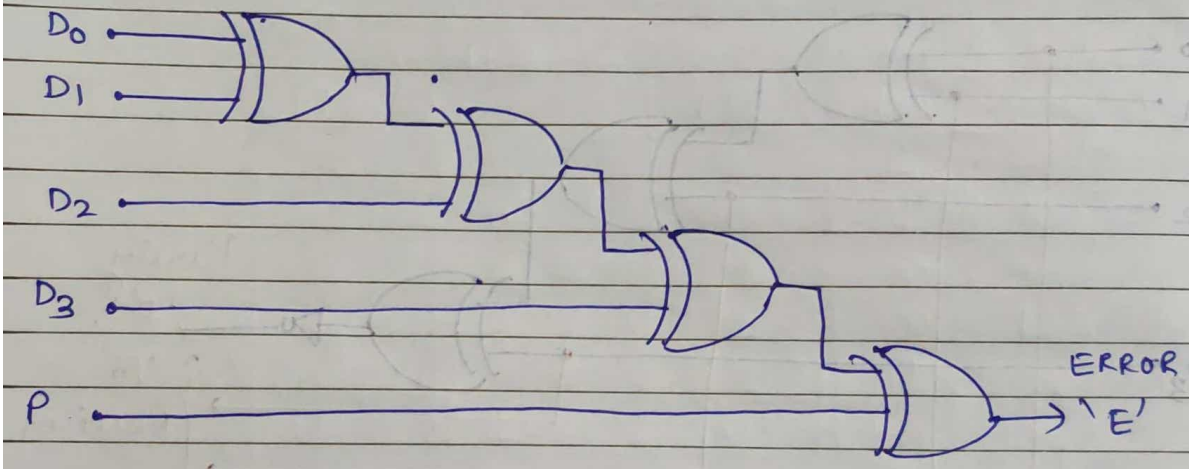
(b) Odd parity generator

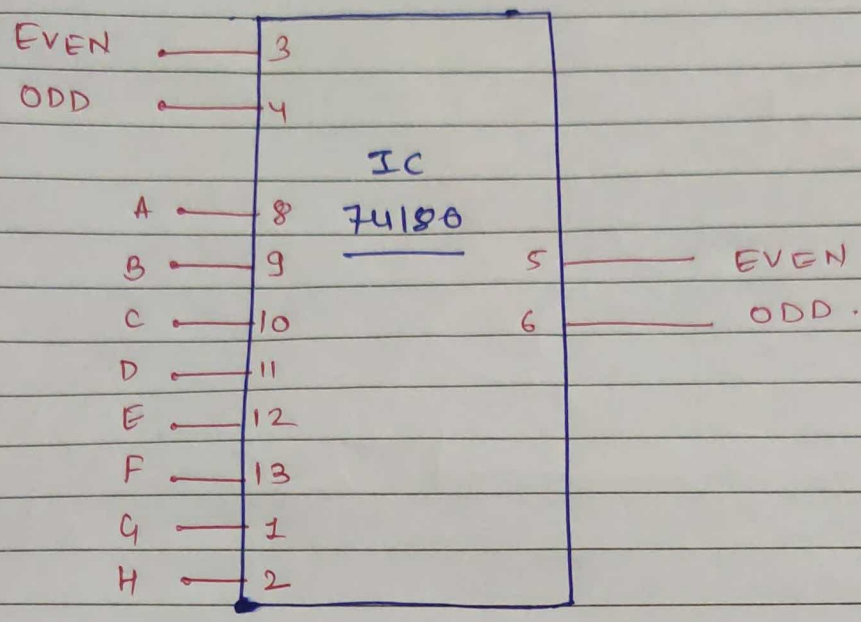


(c) Even parity checker :- (Receiver)



(D) ODD parity checker.





Truth Table :-

Summation of 1's at A → H.	Input		Output	
	Even	Odd	Even	Odd
even	1	0	1	0
odd	1	0	0	1
even	0	1	0	1
Odd	0	1	1	0
x	1	1	0	0
x	0	0	1	1