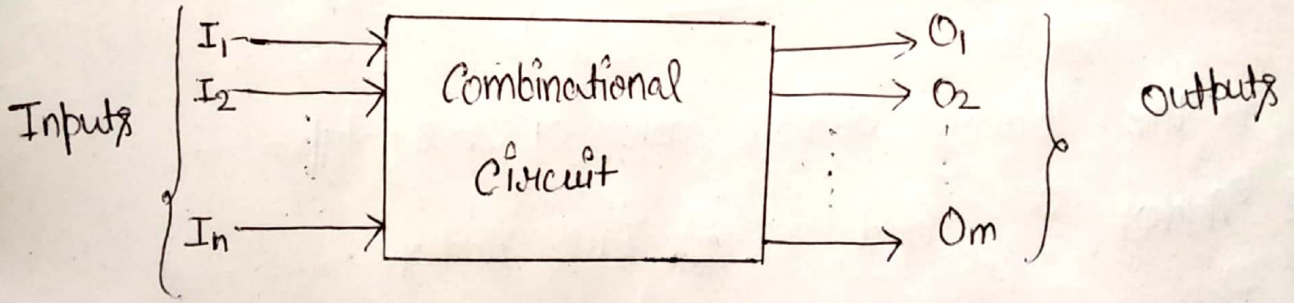
Digital Electronics

BCA - 2nd Sem

Combinational Logic Circuits

Combinational Circuit :- Combinational logic circuits are memoryless digital logic circuits whose output at any instant in time depends only on the combination of its inputs.

A combinational circuit can have a number of inputs and a number of outputs. The block diagram of combination circuit having n inputs and m outputs are as -



The sequence in which the inputs are being applied has no effect on the output of a combinational circuit.

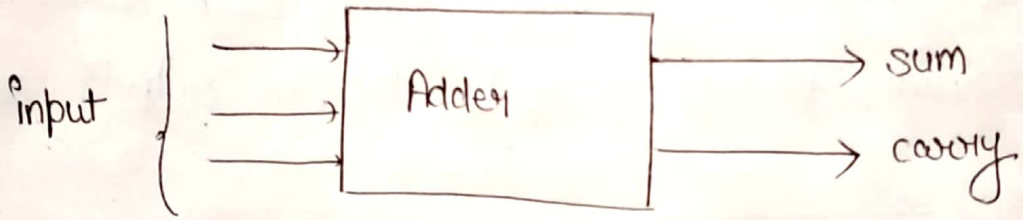
Example - Following are the examples of some combinational circuits -

Adders subtractors, multiplexers, demultiplexers, decoder, encoders etc.

Note – The combinational circuits do not use any memory because it works only on current input.

Binary Adder

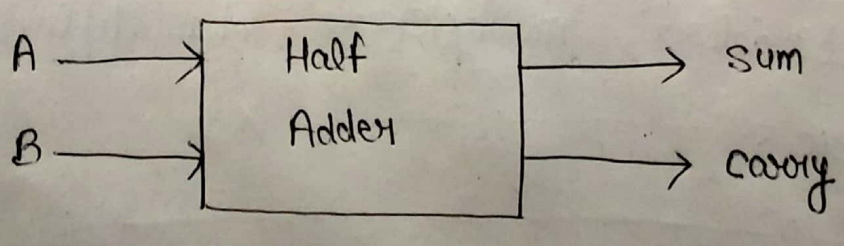
Binary adder is a combinational circuit that performs addition of binary numbers. It takes two or more binary bits as an input and produces sum and carry as output.



The binary adders are of two types –

1) Half Adder and 2) Full Adder

1) **Half Adder :–** Half adder is a combinational logic circuit that accepts two inputs in form of binary bit and produces two outputs namely 'sum' and 'carry'. Block diagram of half adder is as shown -



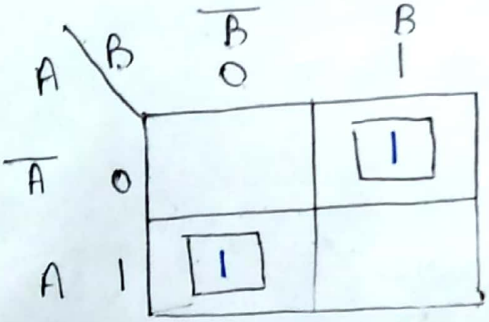
Truth table of half adder is —

Input		output	
A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Now, K-map and simplified expression for outputs —

K-map for sum — Total input variables = 2

∴ No. of blocks = $2^2 = 4$

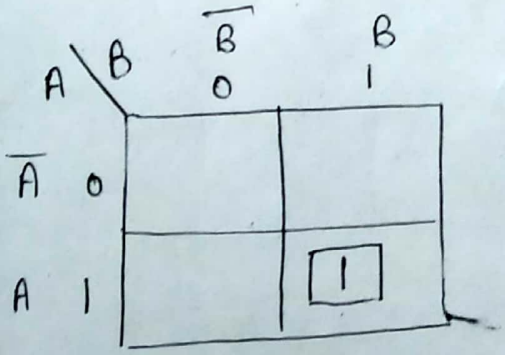


sum = $\overline{A}B + A\overline{B}$

Sum = $A \oplus B$

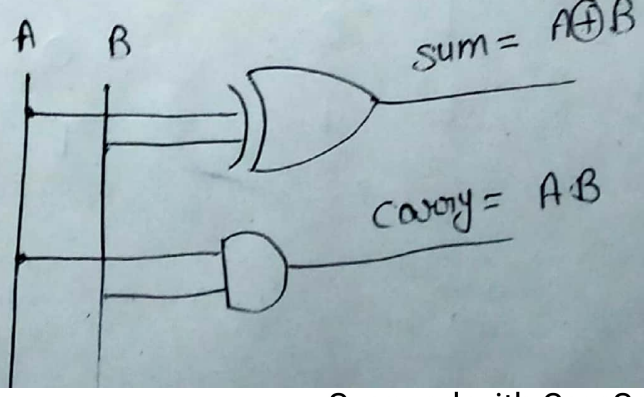
K-map for carry - Two input is used

∴ No of cells = $2^2 = 4$



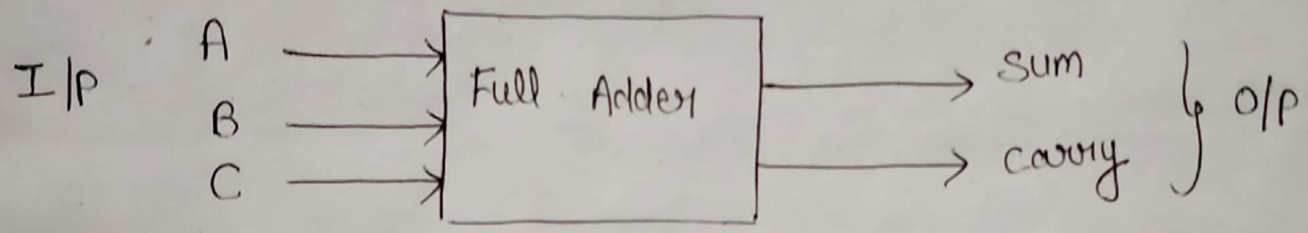
Carry = AB

Logic Diagram —



Full Adder:- To overcome the drawback of half adder circuit, a single bit adder circuit called full adder is developed.

That means a combinational circuit which accept three binary bits (0 & 1) as an input and generate sum & carry as an output. Block diagram are as –



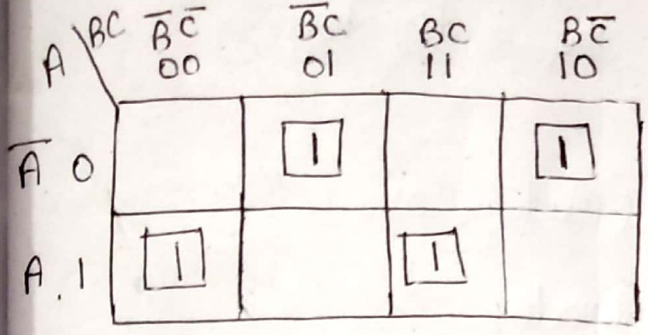
Truth table of Full Adder is given as –

Input			output	
A	B	C	sum	carry
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	1

K map for the sum output —

∵ Total no of variables = 3

∴ Total no of blocks = $2^3 = 8$



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

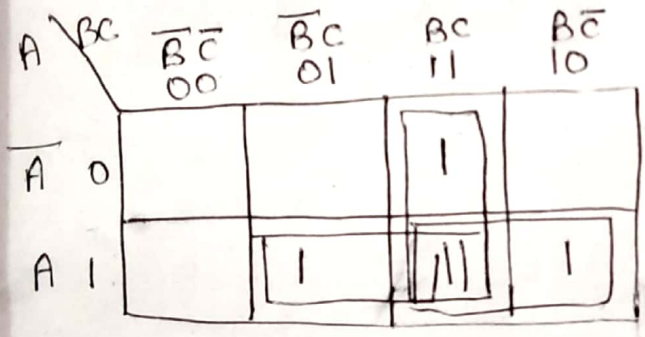
$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$

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

$= A \oplus B \oplus C$

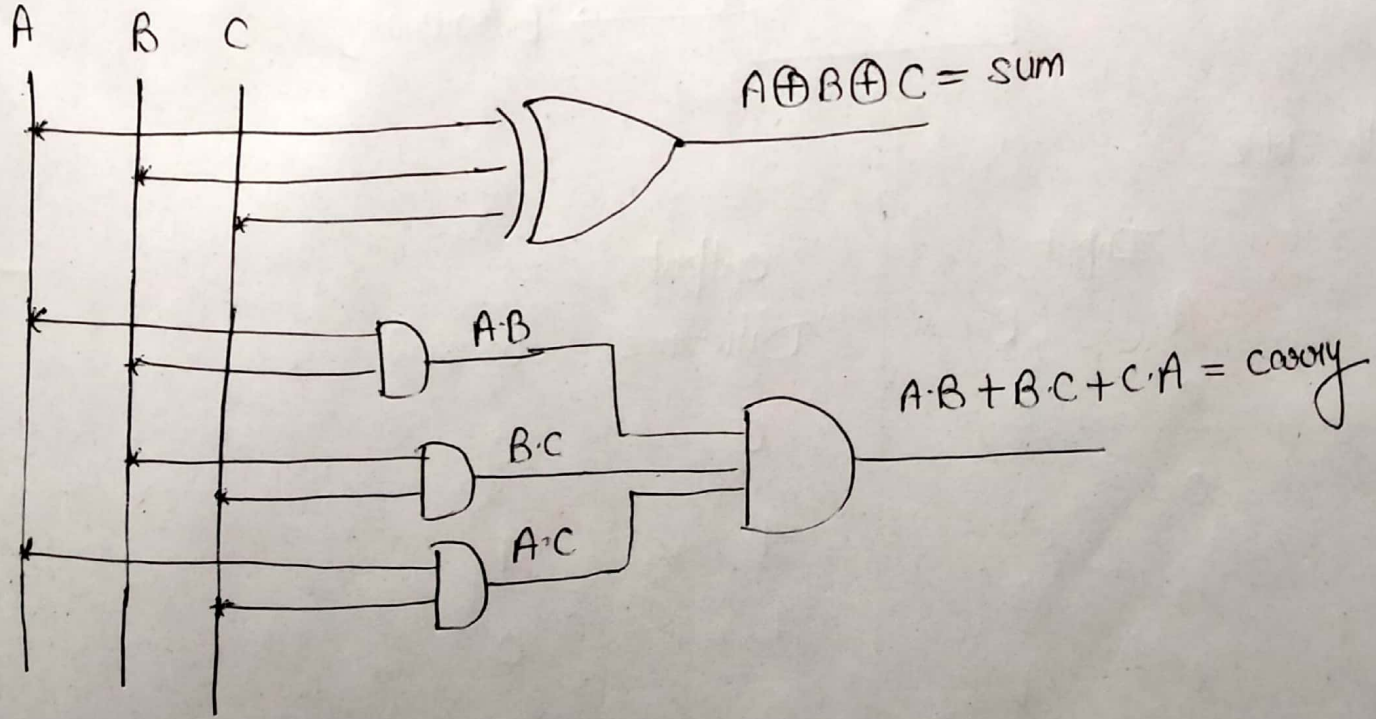
$(\because \bar{A}X + A\bar{X} = A \oplus X)$

K map for Carry output -



Carry = $AB + BC + CA$

Logic Diagram



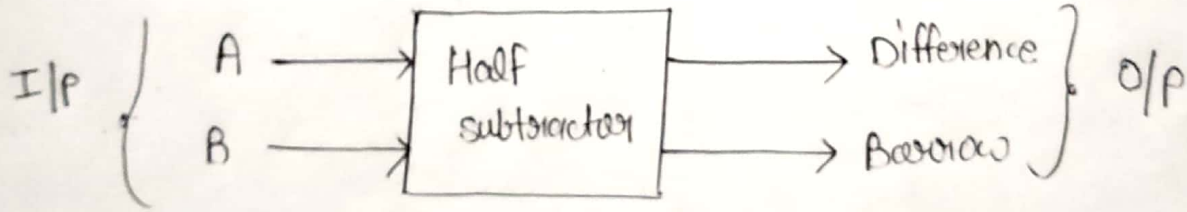
Binary Subtractors :- The binary subtractor is another type of combinational arithmetic circuit that produces an output which is the subtraction of two binary numbers.

The types of binary subtractors are –

1) Half subtractor 2) Full subtractor

1) **Half Subtractor :-** Half subtractors is a combinational circuit with two inputs and produces two outputs difference and borrow.

Block diagram –



Truth Table –

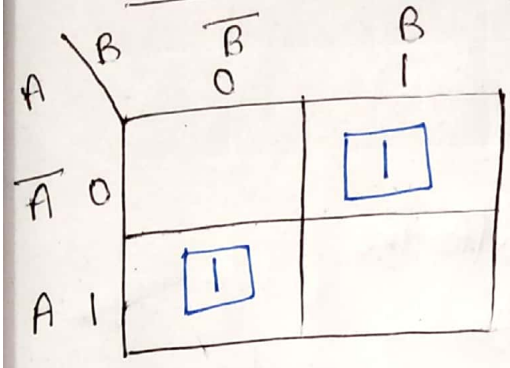
Input		output	
A	B	Difference	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K map for difference and borrow :-

∵ Total number of Input = 2

∴ No of blocks = $2^2 = 4$

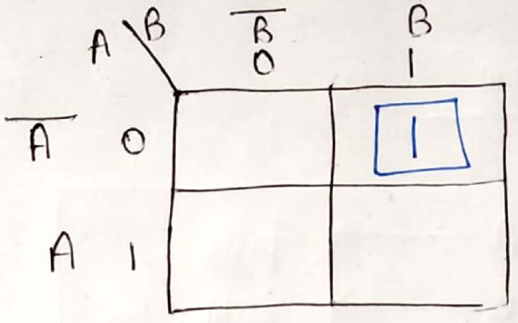
For Difference



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

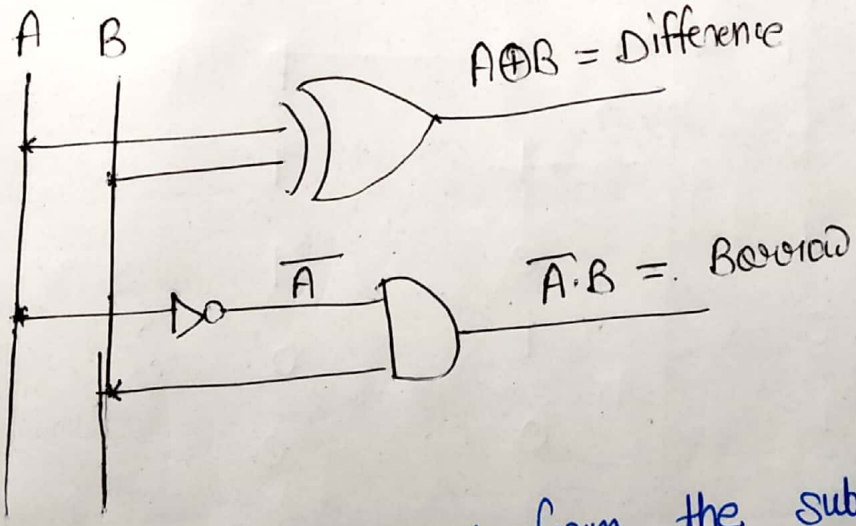
Difference = $A \oplus B$

For Borrow



borrow = $\overline{A}B$

Logic Diagram -

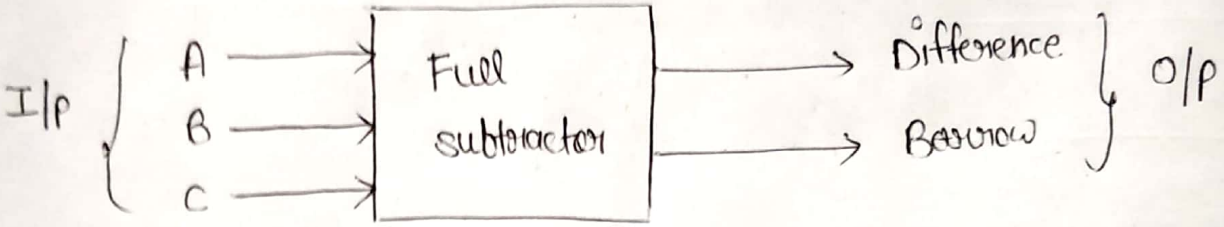


Drawback :- It can only perform the subtraction of two binary bits, but while performing the subtraction, it does not take into account the borrow of the previous significant stage.

Full Subtractor :- The full subtractor is a combination circuit with three inputs and two outputs which is difference and borrow.

The full subtractor is used to overcome the disadvantage of half subtractor.

Block Diagram -



Truth Table -

Inputs			outputs	
A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-map and simplification for both output -

$\because$ Number of inputs = 3

$\therefore$ Number of blocks = $2^3 = 8$

For Difference

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

For borrow

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

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

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

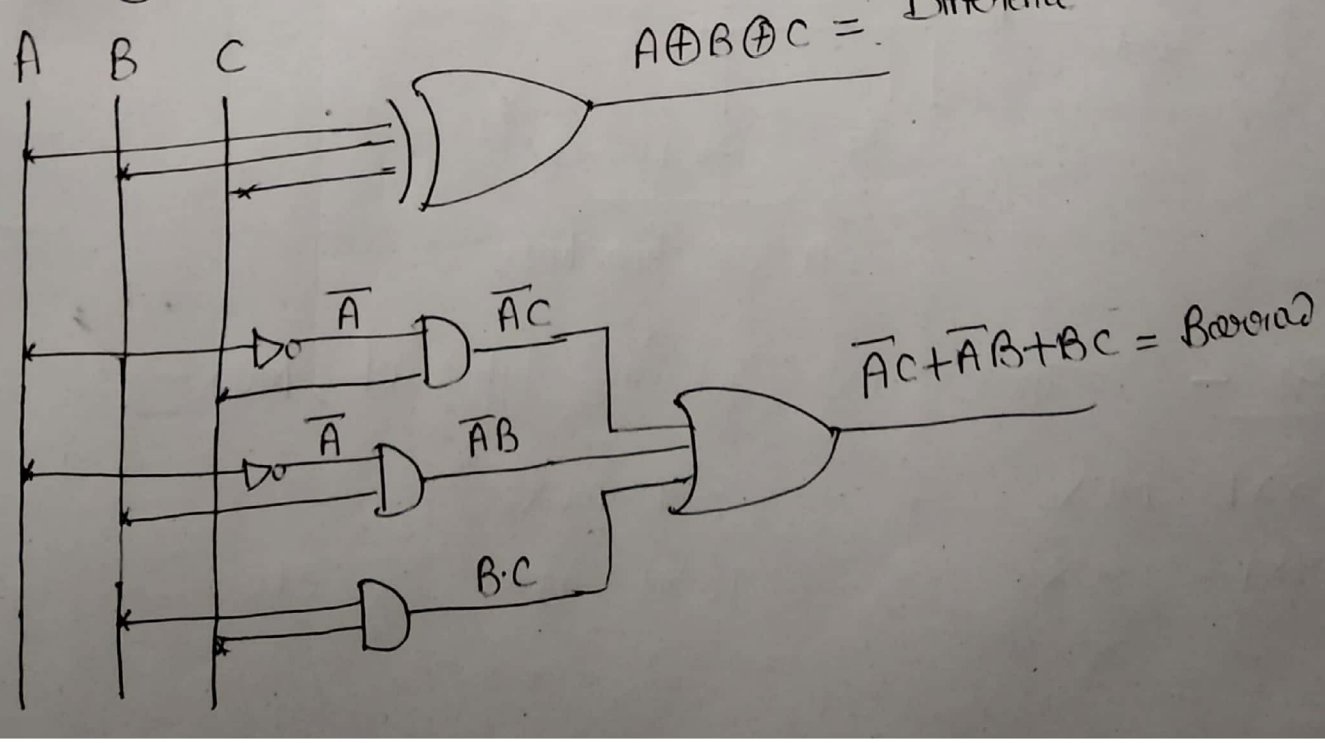
$= \bar{A}(B \oplus C) + A\overline{(B \oplus C)}$

Difference = $A \oplus B \oplus C$

$(\because \bar{A}x + A\bar{x} = A \oplus x)$

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

Logic Diagram -



Multiplexer :- Multiplexer is a special type of combinational circuit. It selects one of the data inputs and routes it to the output with the help of selection lines.

The Multiplexer, shortened to 'MUX' is also called data selector.

The standard form of multiplexer or MUX is $2^n \times 1$ where

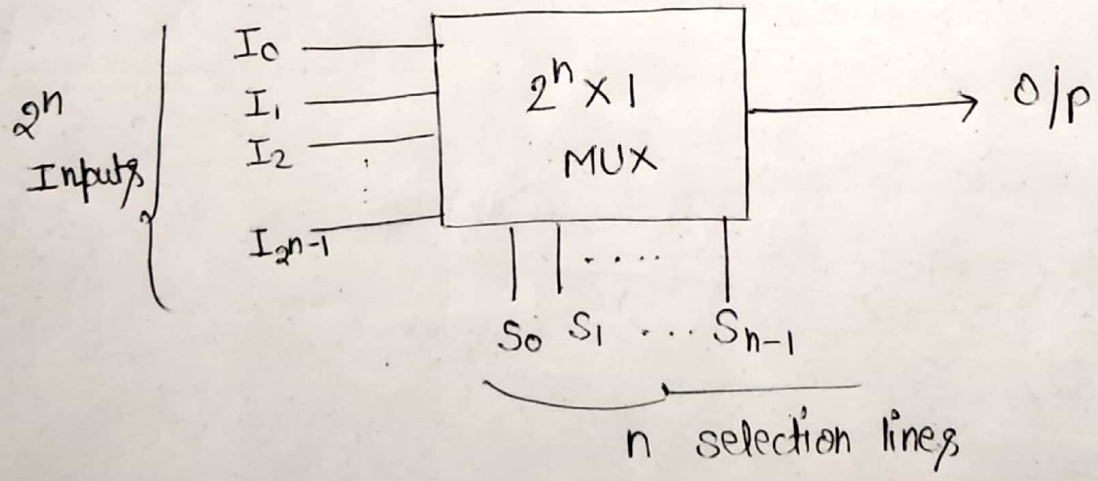
2^n = number of input lines

n = no of selection lines

1 = output lines

& $n = 1, 2, 3, 4 \ldots\ldots$

The block diagram of $2^n \times 1$ multiplexer is –



Examples – There are following types of MUX –

2×1 MUX , 8×1 MUX

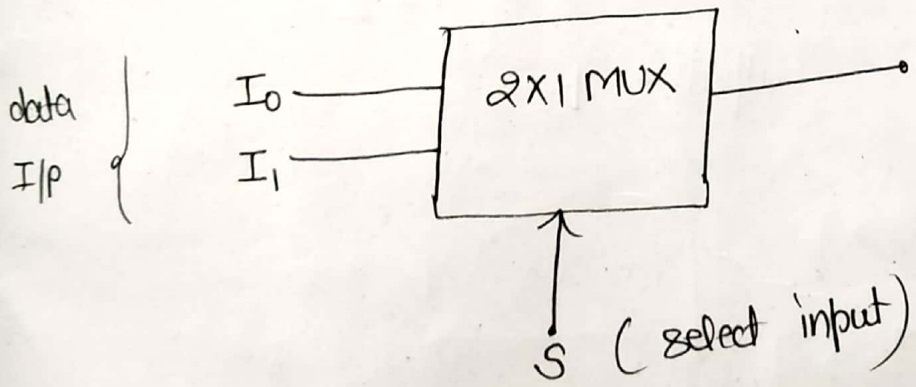
4×1 MUX , 16×1 MUX etc

Advantages of Multiplexer :-

1) It reduces the number of wires, so that it reduces the circuit complexity and cost.
2) It simplifies the logic design.
3) It does not need the K maps and simplification
4) With the help of MUX, we can implement many combinational circuits.

2X1 Multiplexer :- It has two data inputs I_0 & I_1, one select input S and one output Y.

The block diagram of 2X1 MUX is -

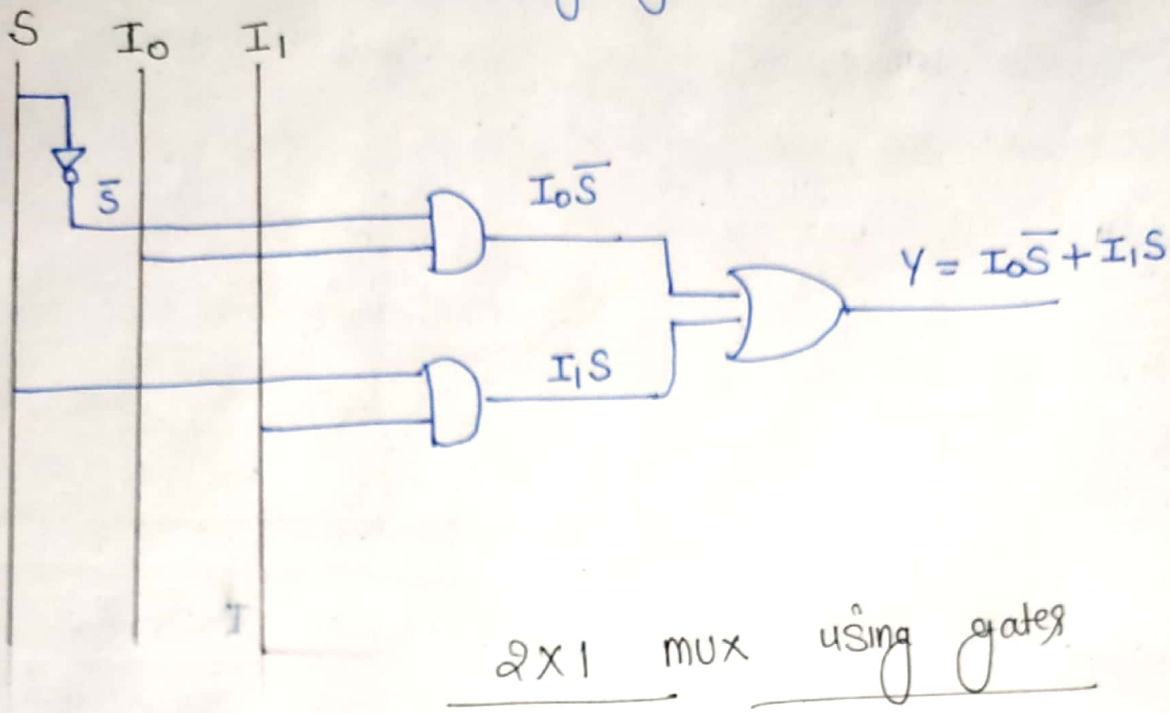


Truth Table -

select input	output
S	Y
0	I_0
1	I_1

$\therefore \quad Y = I_0\overline{S} + I_1S$

The realization using gates is –

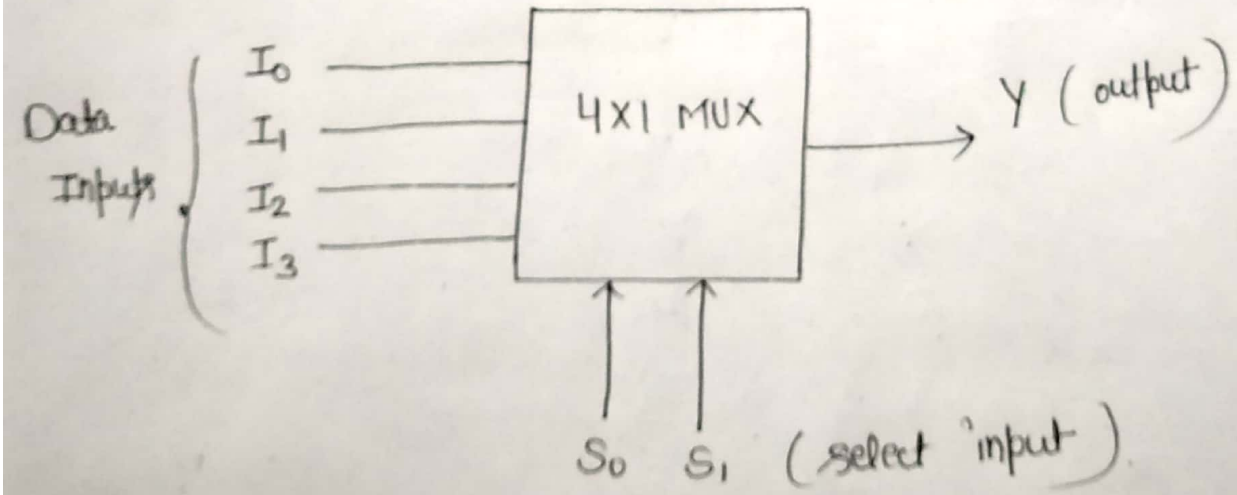


2X1 MUX using gates

4X1 MUX :-

Number of Input lines = 4
Number of selection lines = 2
& output = 1

Block diagram –



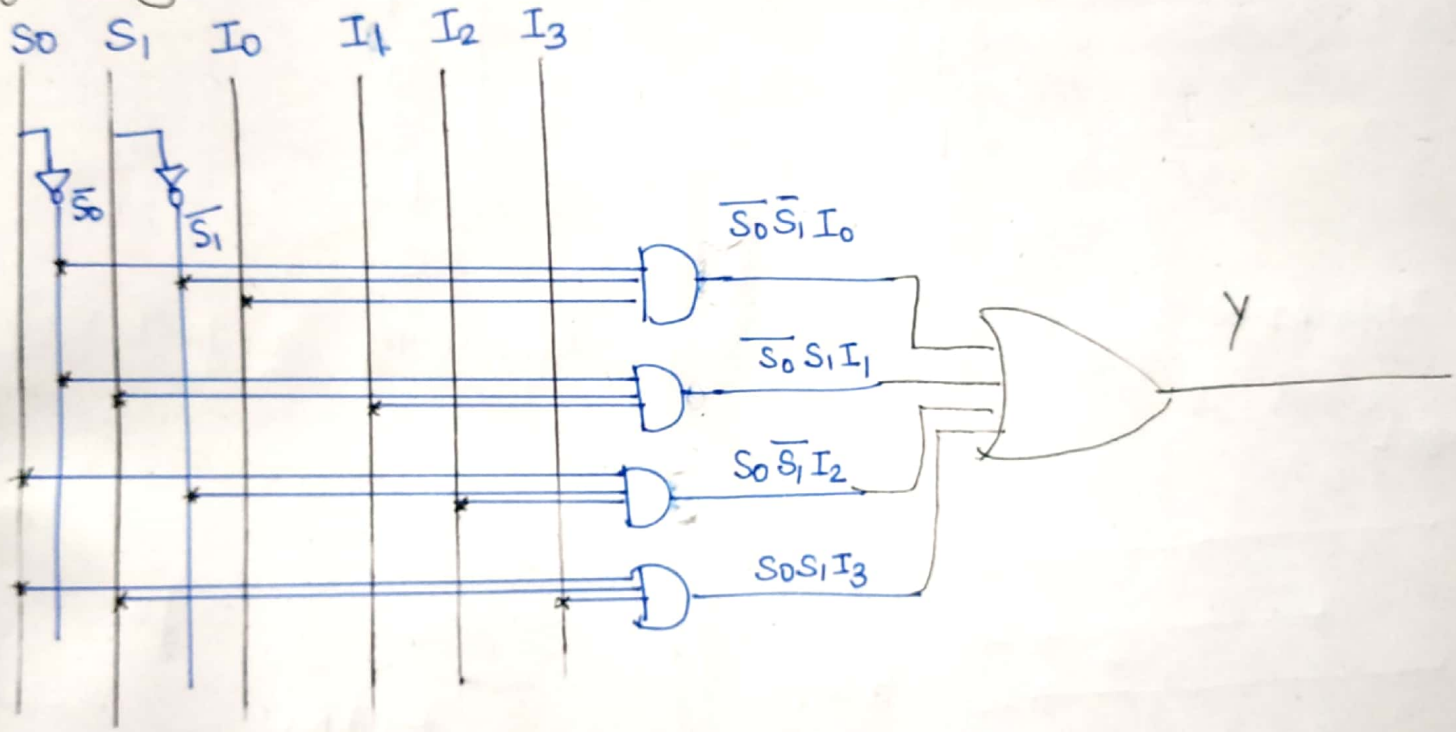
∴ Truth table is –

Select inputs		output
S_0	S_1	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

output is

$$Y = \overline{S_0}\,\overline{S_1} I_0 + \overline{S_0} S_1 I_1 + S_0 \overline{S_1} I_2 + S_0 S_1 I_3$$

Logic diagram -



Example - Implement the following expression using a multiplexer

$$F(A,B,C) = \sum m(0,2,4,6)$$

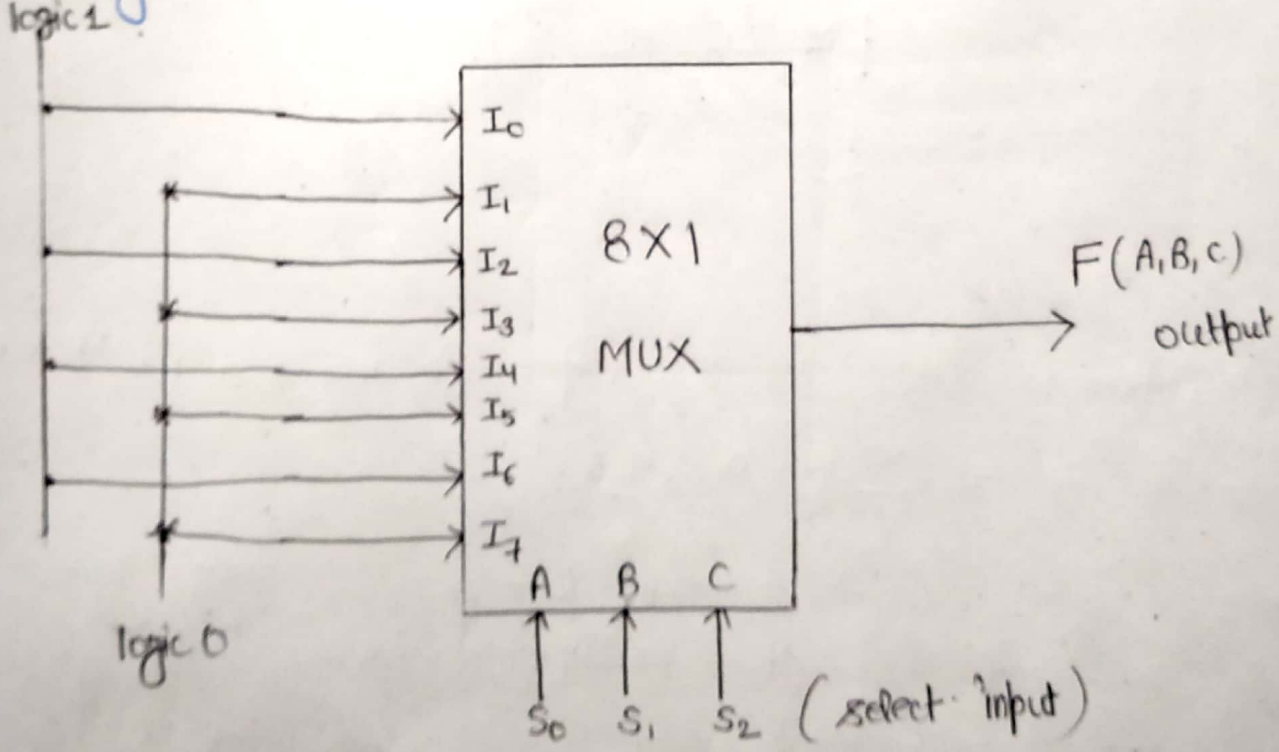
Sol Since, value used between 0-7, it means three variables are used. So, the multiplexer having three select inputs should be used i.e. 8×1 mux used.

In the given expression, connect the data input 0, 2, 4, 6 to logic 1 and remaining to logic 0.

Truth Table is –

select inputs			output
A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Implementation of a logic expression using a multiplexer is defined as –



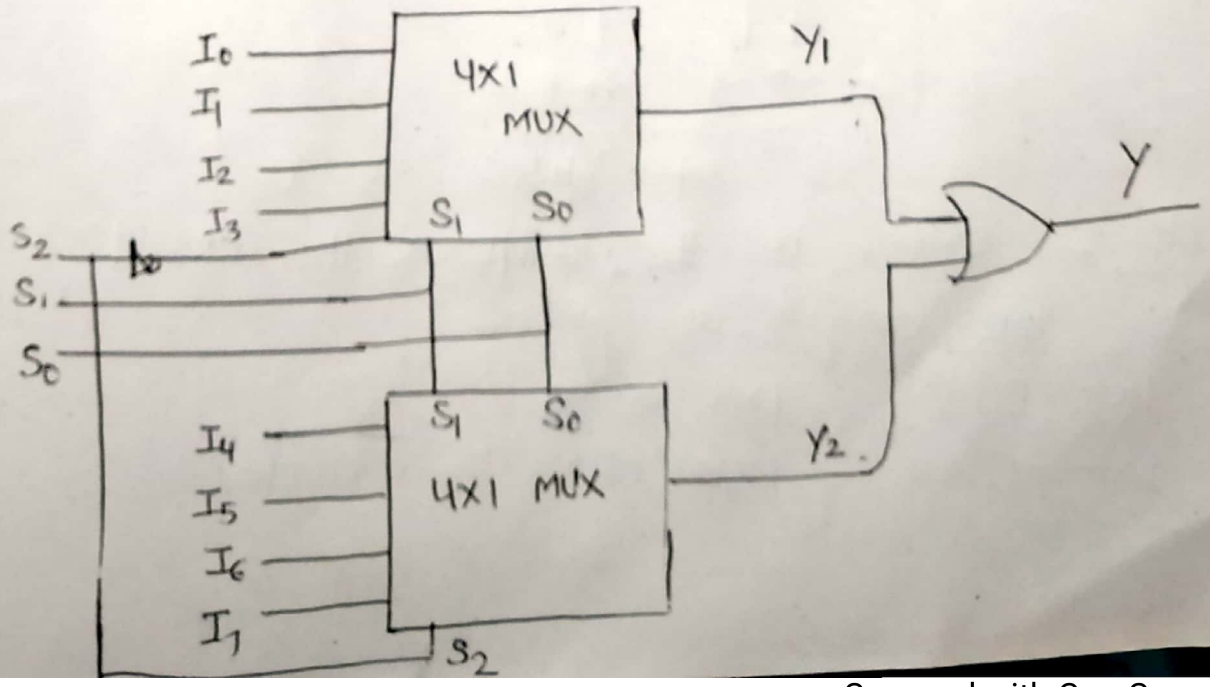
Example - Obtain an 8×1 multiplexer using two 4×1 MUX.

Sol The cascading of two 4×1 MUX, there is 8 data inputs (I_0 through I_7) and select lines S_0 & S_1 of both 4×1 MUX are connected in parallel whereas a third select input S_2 is used for one multiplexer at a time.

Truth Table -

select inputs			output	
S_2	S_1	S_0	Y	
0	0	0	I_0	MUX 1
0	0	1	I_1	
0	1	0	I_2	
0	1	1	I_3	
1	0	0	I_4	MUX 2
1	0	1	I_5	
1	1	0	I_6	
1	1	1	I_7	

Implementation of 8×1 using two 4×1 MUX



Demultiplexer :- A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs. At a time only one output line is selected by the select lines and the input is transmitted to the selected output line.

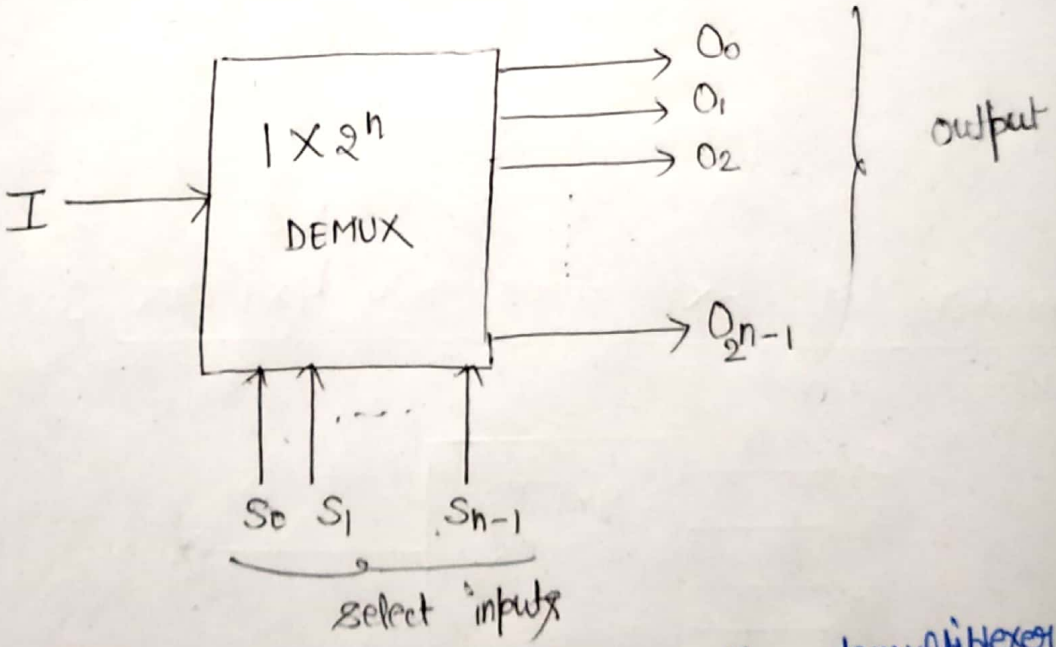
The standard form of demultiplexer or DEMUX is 1×2^n where $n = 1, 2, \ldots$

No of input lines = 1

No of output lines = 2^n

No of selection lines = n

The block diagram of a demultiplexer is shown as -

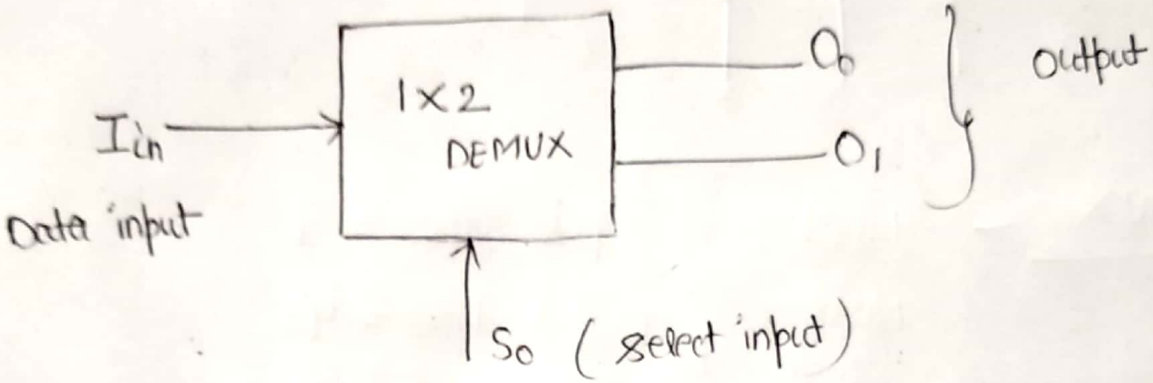


Example - Similar to the multiplexer, the demultiplexer are classified as follows -

1X2 DEMUX, 1X4 DEMUX

1X8 DEMUX, 1X16 DEMUX ---- etc

1×2 Demultiplexer :- 1×2 DEMUX has one data input Iin, one select input So and two output O_0 and O_1. The block diagram of 1×2 DEMUX is shown as –



Truth Table –

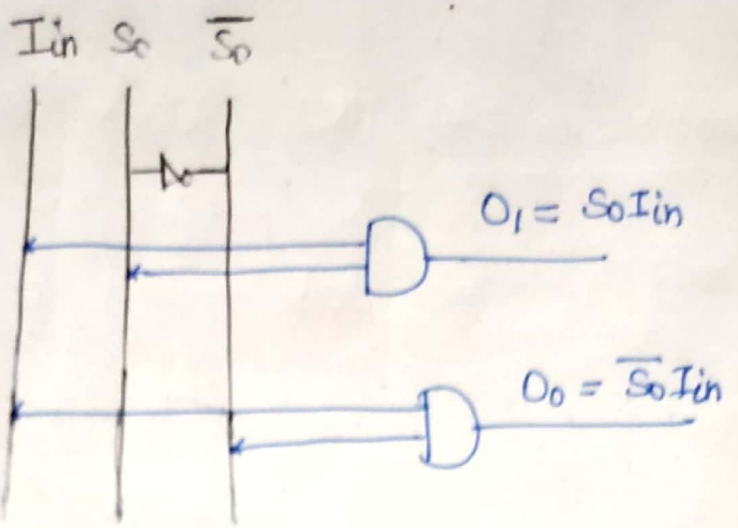
select input	output	
So	O_0	O_1
0	Iin	0
1	0	Iin

Detail truth table is –

input		output	
So	Iin	O_1	O_0
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

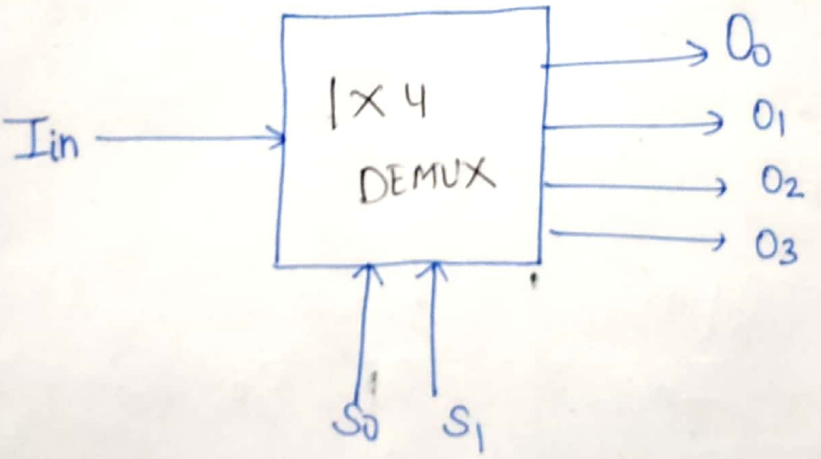
$\therefore \quad O_1 = S_o I_{in} \quad \& \quad O_0 = \overline{S_o} I_{in}$

Logic diagram -



1X4 DEMUX – Number of input lines = 1
Number of output lines = 4
& Number of selection lines = 2

Block diagram –

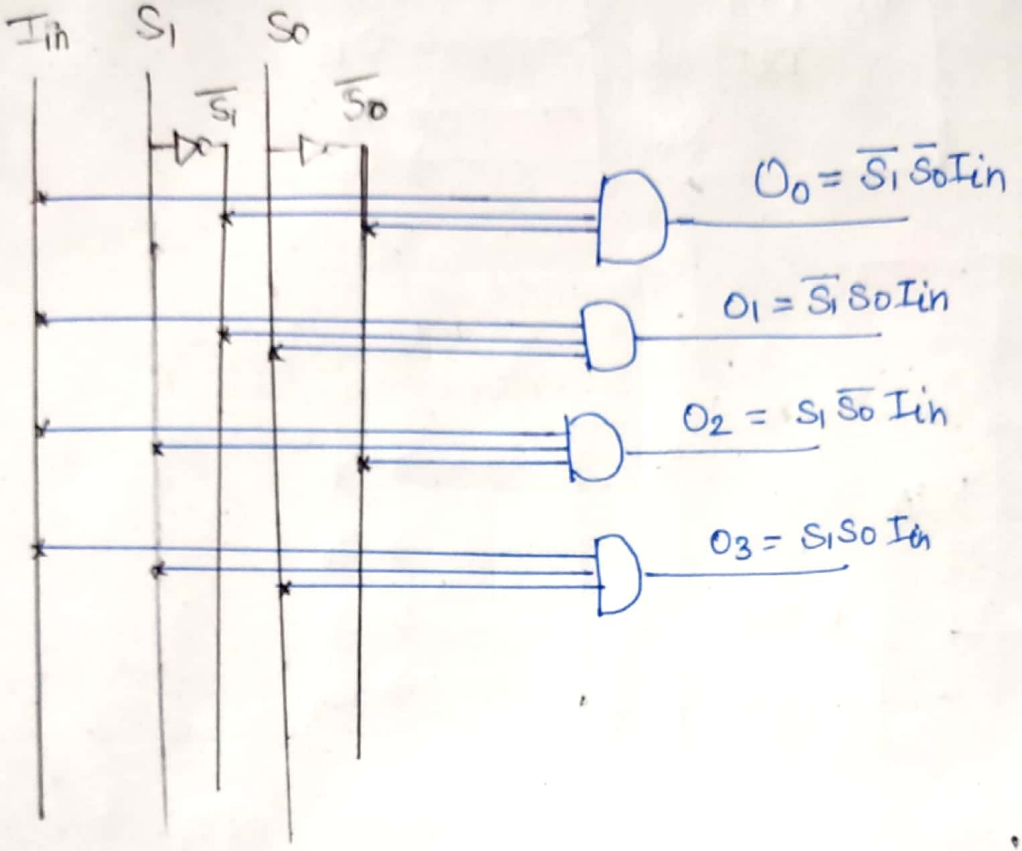


Truth table -

	Input		output			
Iin	S_1	S_0	O_3	O_2	O_1	O_0
Iin	0	0	0	0	0	Iin
Iin	0	1	0	0	Iin	0
Iin	1	0	0	Iin	0	0
Iin	1	1	Iin	0	0	0

$\therefore \quad O_0 = \overline{S_1}\,\overline{S_0}\,I_{in}, \quad O_1 = \overline{S_1}\,S_0 I_{in}$

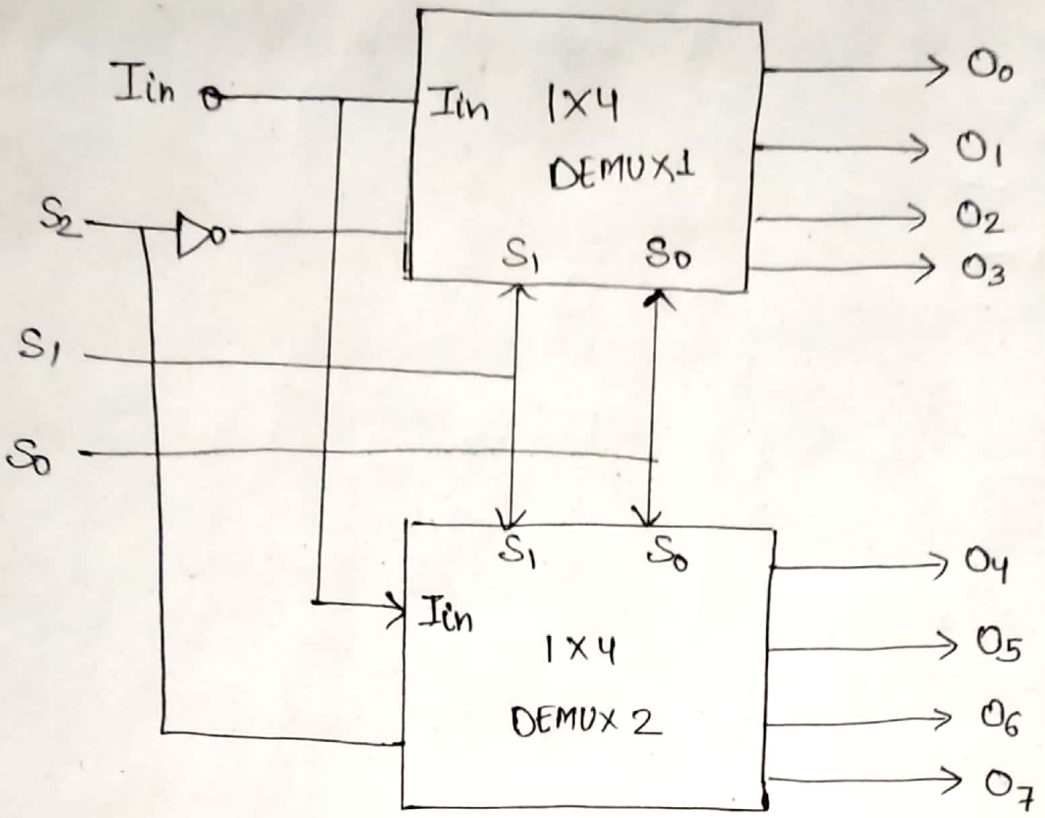
$O_2 = S_1\,\overline{S_0}\,I_{in} \quad \& \quad O_3 = S_1 S_0 I_{in}$



Example – Obtain a 1×8 demultiplexer using two 1×4 line demultiplexer.

Sol The cascading of two 1×4 DEMUX, there is 1 data inputs and 8 output. The select lines S_1 and S_0 are connected in parallel to both 1×4 DEMUX for selecting one of the two 1×4 DEMUX whereas S_2 select line is connected directly to the input of demux 2 ~~where~~ while inverted S_2 is connected to the demux 1

Block diagram are shown as –



Truth Table –

Select inputs			outputs								
S_2	S_1	S_0	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	
0	0	0	1	0	0	0	0	0	0	0	Dem 1
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	Dem 2
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	

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

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

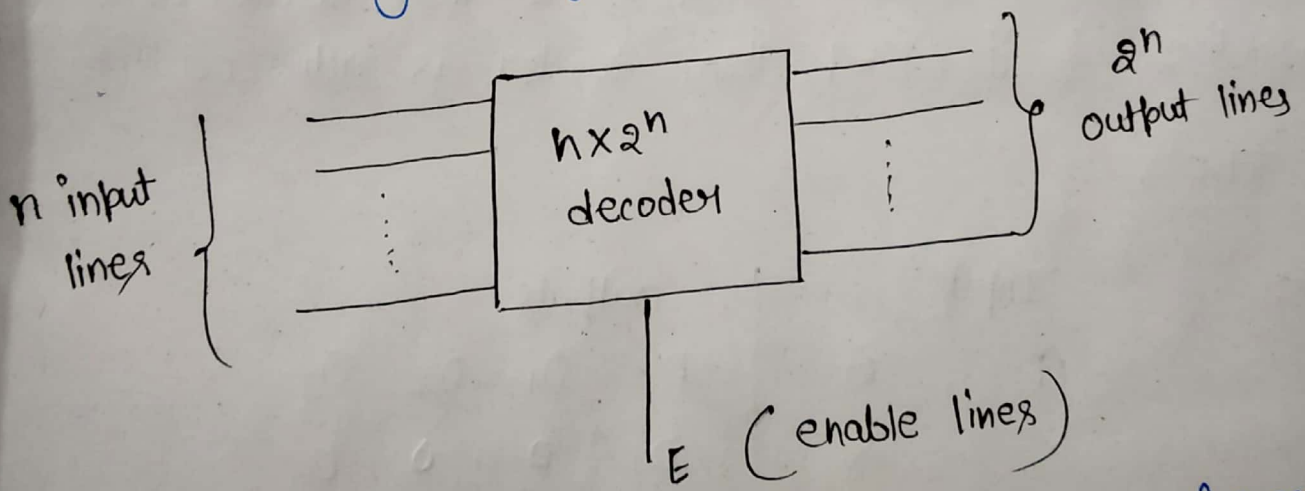
The standard form of decoder is $n \times 2^n$ where

$(n = 1, 2, 3, \ldots\ldots)$

n = Number of input lines

2^n = Number of output lines

Block diagram of decoder is shown as—



Example— Various types of decoder is as follows—

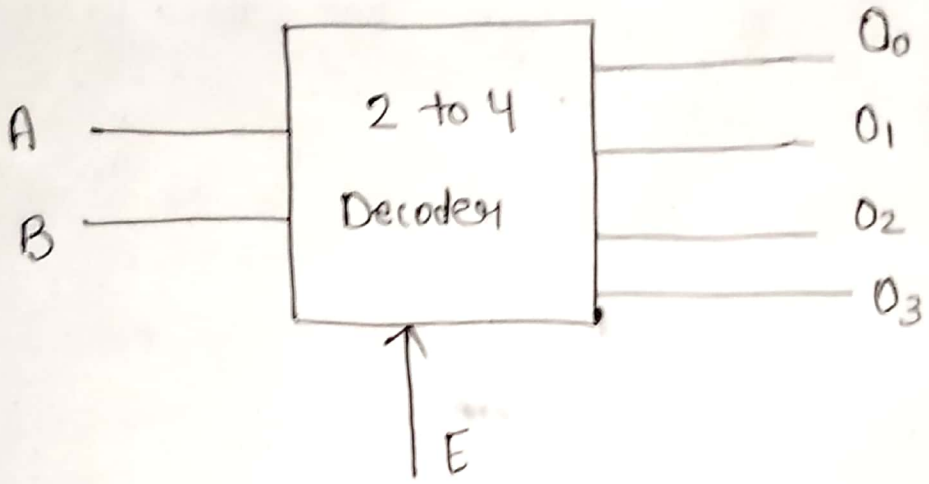
1 X 2 Decoder, 2 X 4 decoder

3 X 8 decoder, 4 X 16 decoder

2×4 Decoder :-

Number of inputs = 2
& number of outputs = 4

Block diagram -



E is called as a strobe or enable input which is useful for cascading. when E = 0 then decoder circuit is deactive & if E = 1 then decoder is active condition.

Truth Table -

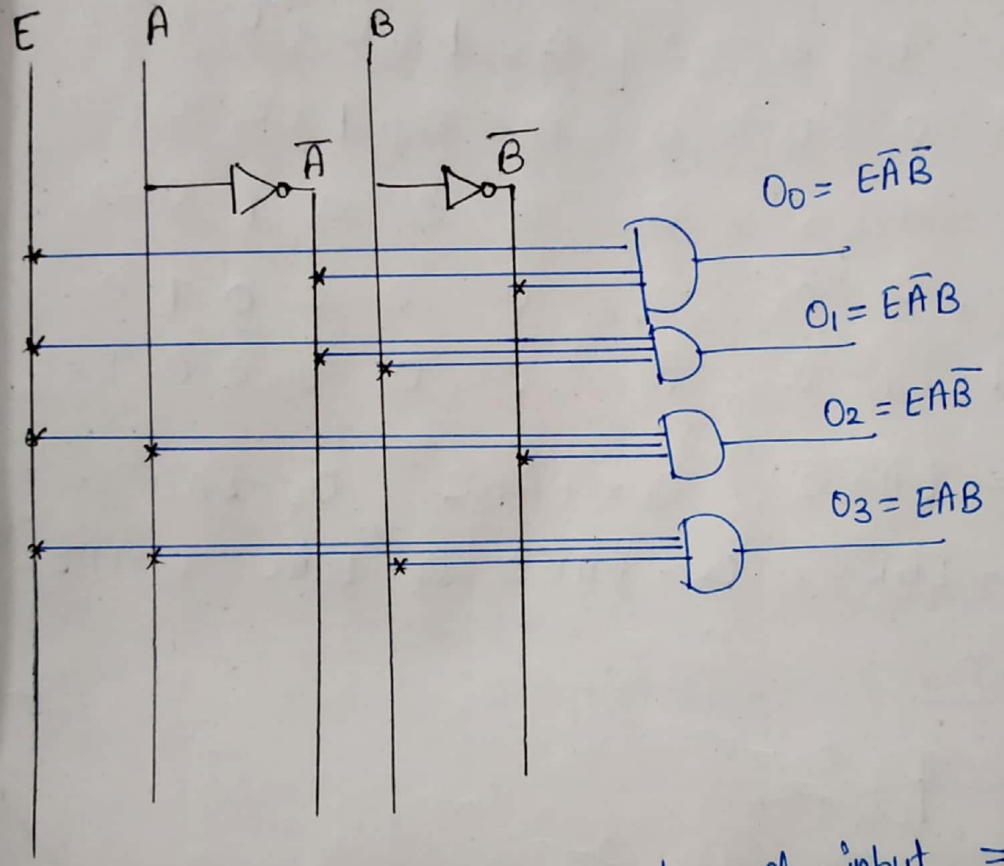
Enable	Inputs		outputs			
E	A	B	D_0	O_1	O_2	O_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

From truth table, the boolean expression for the four outputs are –

$$O_0 = E\bar{A}\bar{B} \quad , \quad O_1 = E\bar{A}B$$

$$O_2 = EA\bar{B} \quad , \quad O_3 = EAB$$

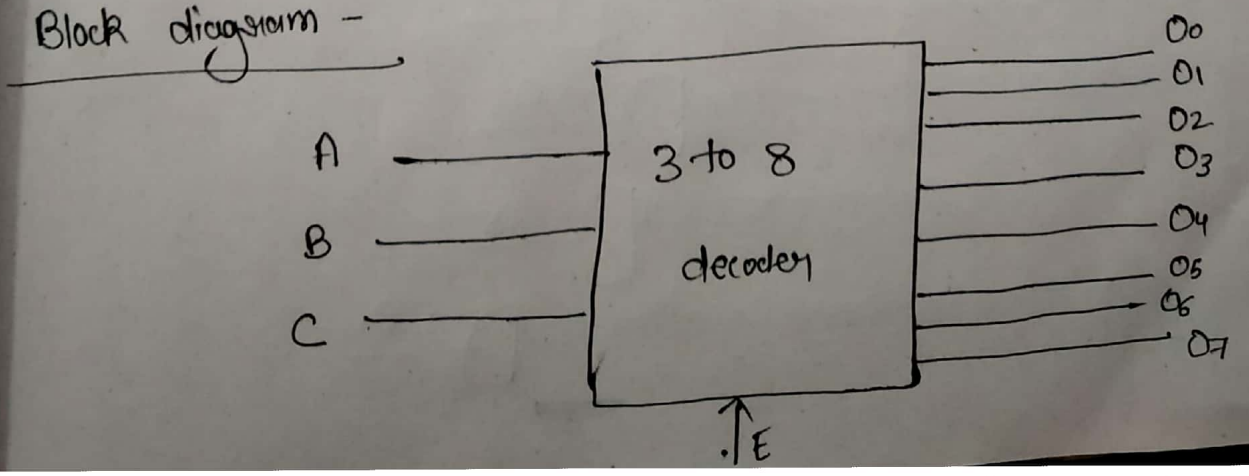
The circuit diagram of 2 to 4 decoder is shown in the following figure –



3 to 8 line decoder –

number of input = 3
number of output = 8

Block diagram –

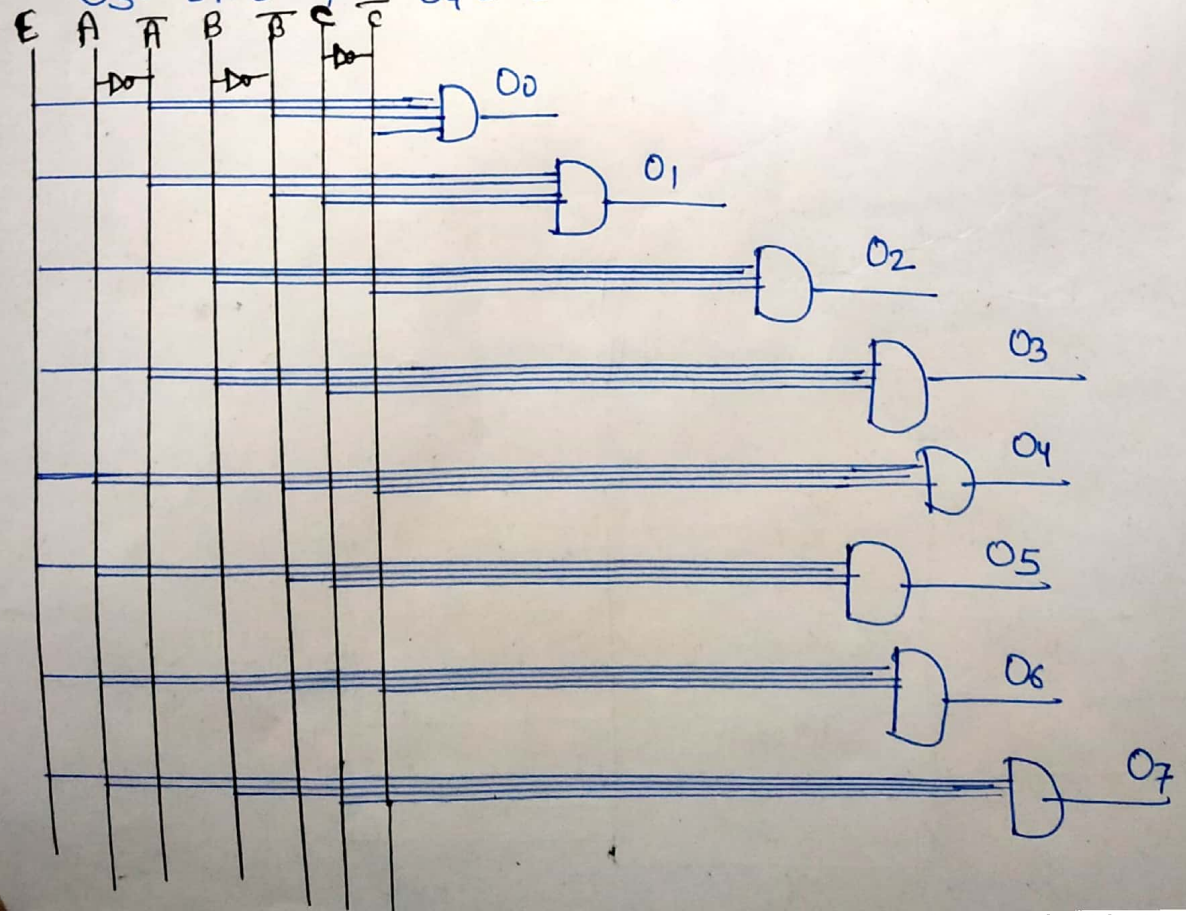


Truth table –

Enable	Input			output							
E	A	B	C	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

output is – $O_0 = E\bar{A}\bar{B}\bar{C}$, $O_1 = E\bar{A}\bar{B}C$, $O_2 = E\bar{A}B\bar{C}$

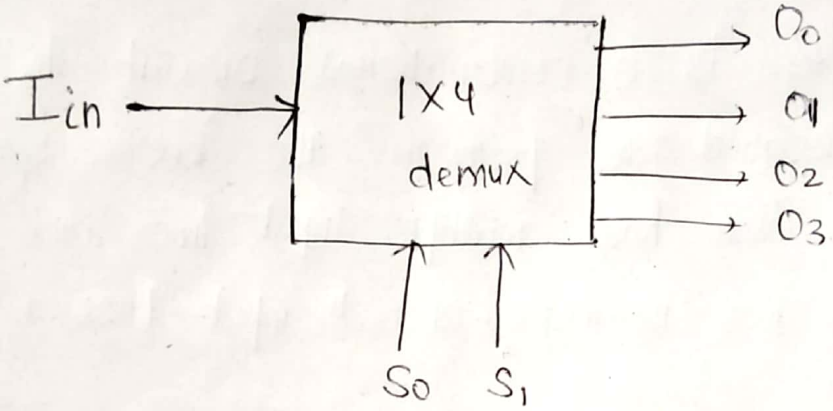
$O_3 = E\bar{A}BC$, $O_4 = EA\bar{B}\bar{C}$, $O_5 = EA\bar{B}C$, $O_6 = EAB\bar{C}$, $O_7 = EABC$



Demultiplexer as Decoder :-

It is possible to operate a demultiplexer as a decoder. Now, we operate a 1×4 demultiplxer as 2×4 decoder.

Consider 1×4 demux as—



Here Iin is the data input, S_0, S_1 are the selection lines and O_0 through O_3 are the outputs.

In order to operate it as 2×4 decoder, we have to use S_1, S_0 as input, keep Iin as enable as open and use O_0 through O_3 as output.

Truth table :-

enable	Inputs		outputs				
Iin (E)	S_0	S_1	O_0	O_1	O_2	O_3	
1	0	0	1	0	0	0	$E\bar{S_0}\bar{S_1}$
1	0	1	0	1	0	0	$E\bar{S_0}S_1$
1	1	0	0	0	1	0	$ES_0\bar{S_1}$
1	1	1	0	0	0	1	ES_0S_1

Applications of Decoder :-

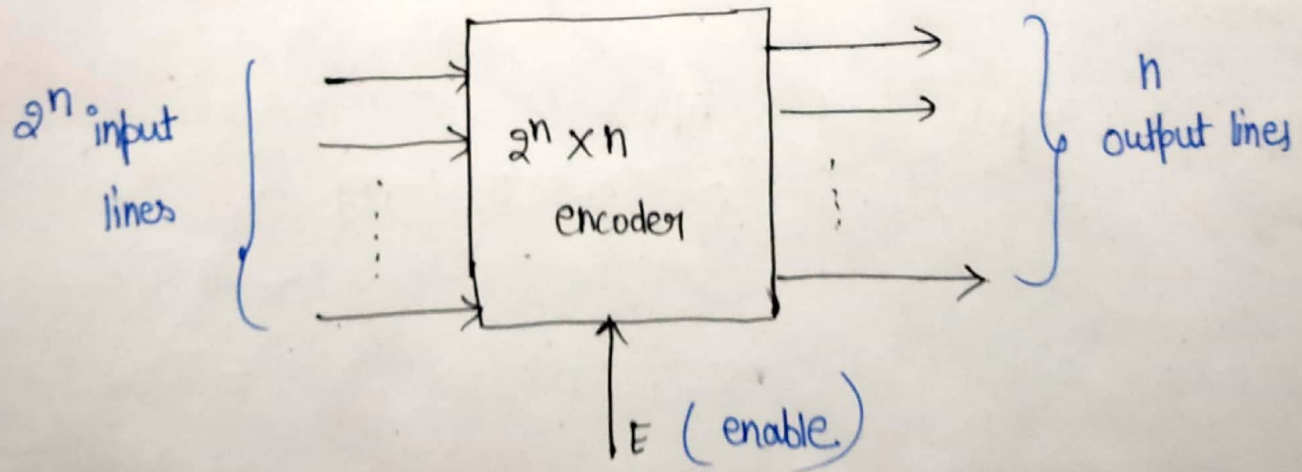
- Code converters
- BCD to seven segment Display
- Relay actuators

Encoder :- Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder. It is also have multiple input lines and multiple output lines. It has maximum of 2^n input lines and n output lines.

The standard form of encoder is $2^n \times n$ ($n = 1, 2, \ldots$) where 2^n = number of input lines

n = number of output lines

Block diagram -



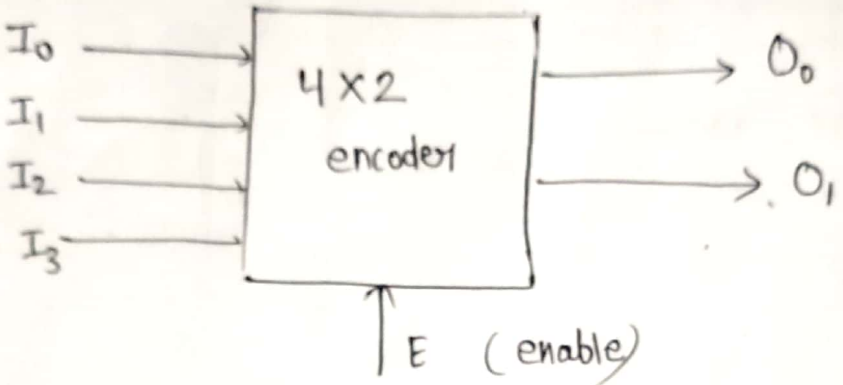
Example - 4×2 encoder, 8×3 encoder, 16×4 encoder ... etc.

4 to 2 encoder :-

Number of input lines = 4

Numbers of output lines = 2

Block diagram -



Truth table -

Inputs				outputs	
I_0	I_1	I_2	I_3	O_0	O_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

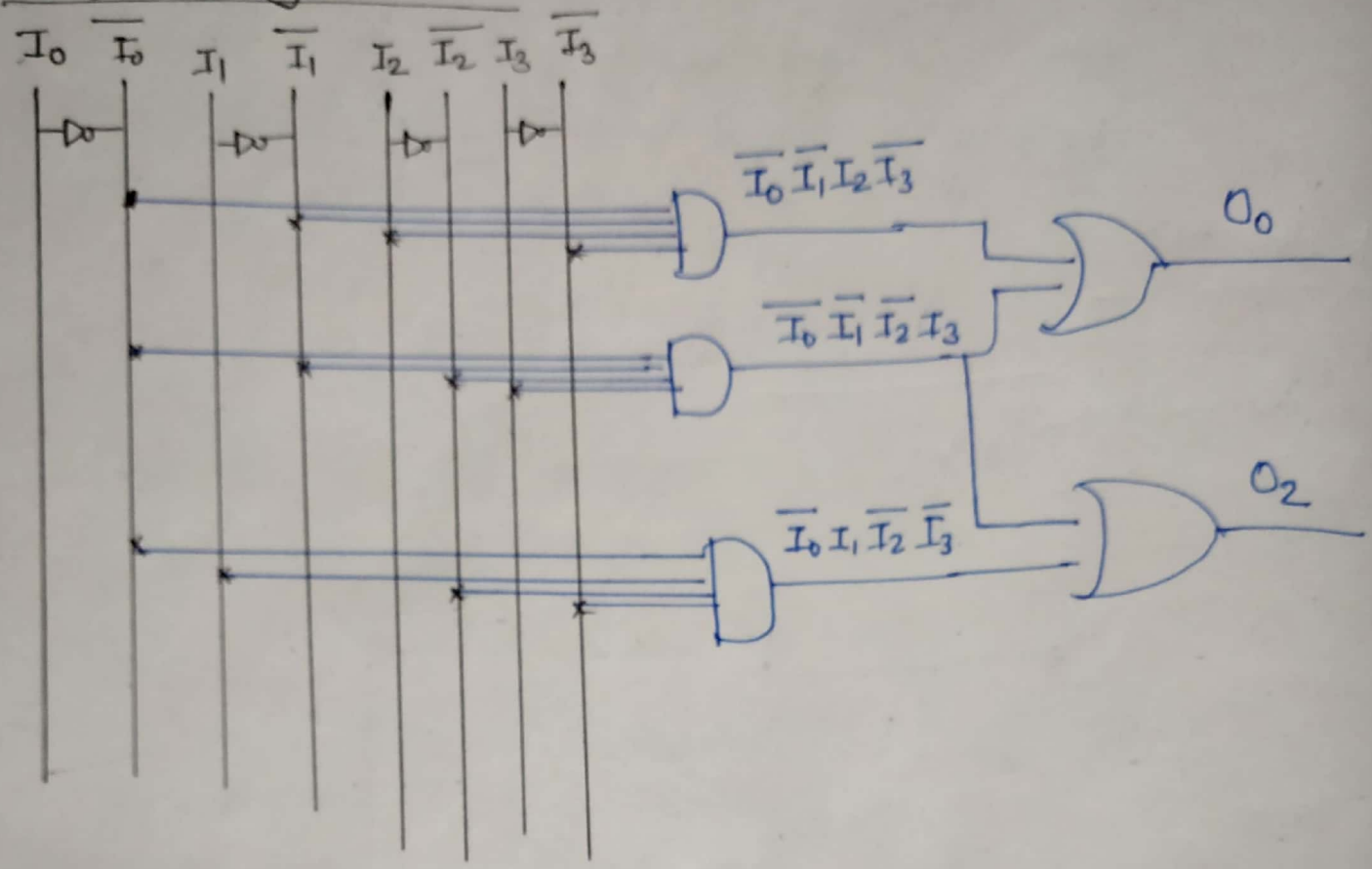
The boolean expression for the output is -

$$O_0 = \overline{I_0}\,\overline{I_1}\,I_2\,\overline{I_3} + \overline{I_0}\,\overline{I_1}\,\overline{I_2}\,I_3$$

&

$$O_1 = \overline{I_0}\,I_1\,\overline{I_2}\,\overline{I_3} + \overline{I_0}\,\overline{I_1}\,\overline{I_2}\,I_3$$

Logic Diagram is—



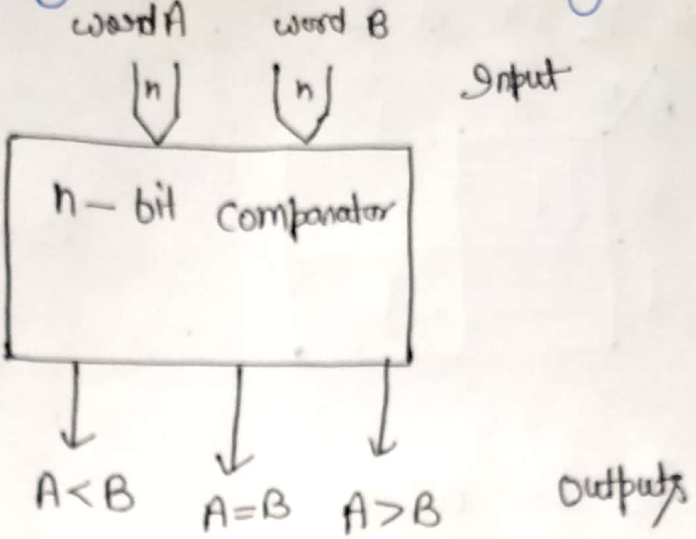
Application of Ecoder is—

- Priority encoder
- Decimal to BCD encoder
- Octal to Binary encoder
- Hexadecimal to Binary encoder

Binary Comparators :— Digital comparator is a, designed to compare the two n-bit binary words applied at its input.

The comparator has three outputs namely $A > B$, $A = B$ and $A < B$. Depending on the result of comparison, one of these output will go high.

The block diagram of an n-bit digital comparator is –



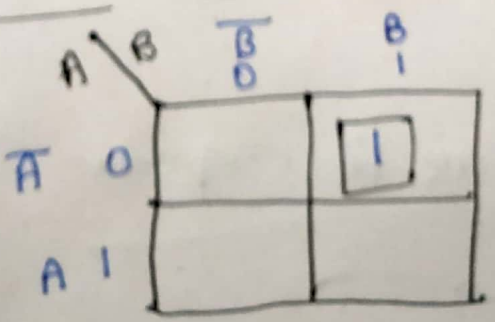
1- bit Binary Comparator :- The one bit comparator is a combinational logic circuit that compares the two single bit numbers A and B and produces an output that indicates the result of the comparison.

Truth Table –

Inputs		outputs		
A	B	$Y_1 = A < B$	$Y_2 = A = B$	$Y_3 = A > B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

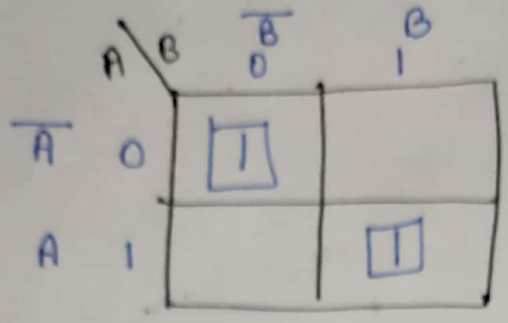
K-map for each output is as – ∵ no of variables = 2
∴ no of blocks = $2^2 = 4$

For $Y_1 = A < B$



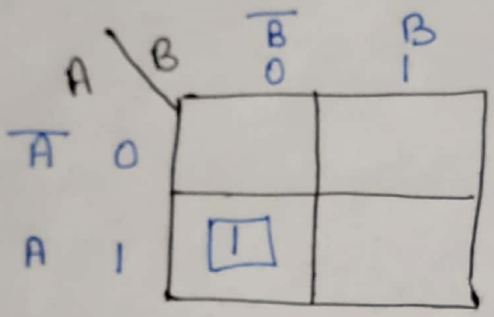
∴ $Y_1 = \bar{A}B$

K-map for Y_2 i.e. $A=B$



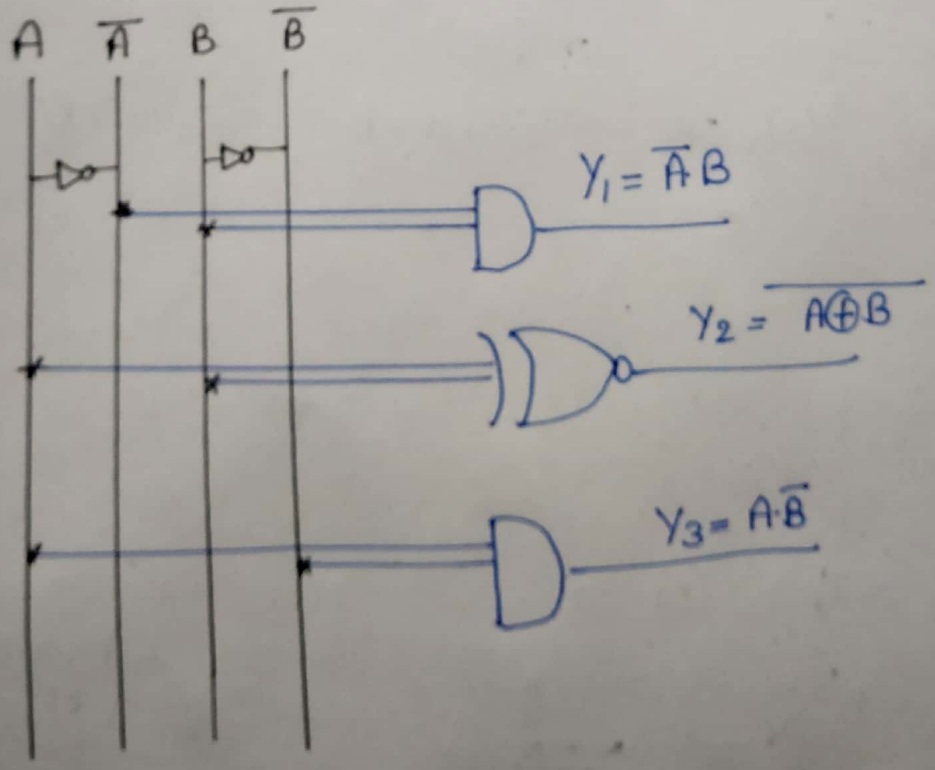
$$Y_2 = \overline{A}\,\overline{B} + AB$$
$$= \overline{A \oplus B}$$

K-map for Y_3 i.e. $A>B$



$$\therefore Y_3 = A\overline{B}$$

Logic Diagram –



one-bit Comparator