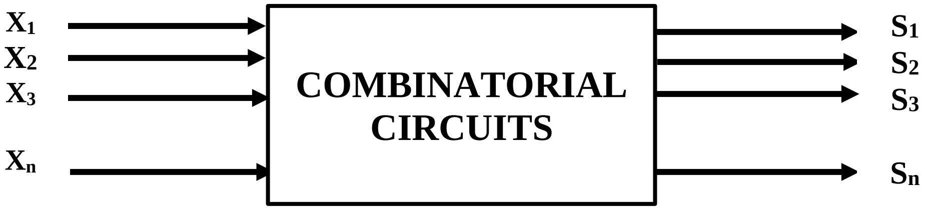
I.INTRODUCTION

Data transmission frequently requires conversion operations. Combinatorial circuits are used for this. To create a combinational logic circuit, several elementary logic gates must be used. There are several combinatorial logic devices commonly used in digital systems. These include encoders, decoders, transcoders, multiplexers, demultiplexers, comparators, etc.

II. DÉFINITION

Combinatorial logic concerns the study of functions whose output value depends only on the logical state of the inputs, resulting in a modification of the output value, and not on its previous states: each combination of input variables always corresponds to a single combination of output functions.



The simple gates we have seen so far can be used to build increasingly complex circuits, right up to the most powerful microprocessors.

We are going to look at some of the most common circuits encountered in Systems Architecture.

III. COMBINATORIAL CIRCUITS

III.1 THE HALF-ADDER :

The half adder is a combinatorial circuit which can perform the arithmetic sum of two numbers A and B on one bit. The result is the sum S and the carry R.

In binary, addition on a single bit is done as follows:

0 + 0 = 00

0 +1 = 01

1 + 0 =01

1 + 1 =10

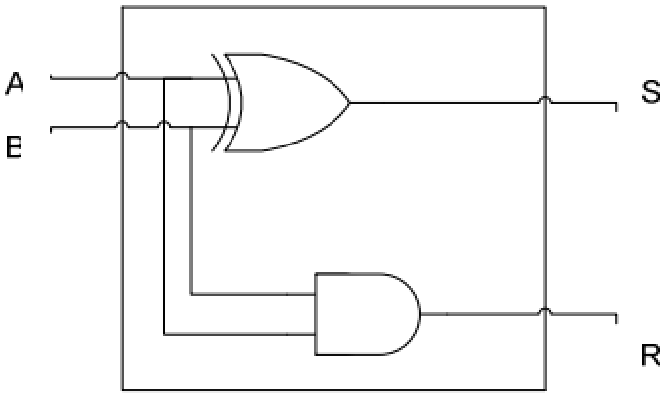
The associated truth table :

A	B		R	S
0	0		0	0
0	1		0	1
1	0		0	1
1	1		1	0

$$R = A.B$$

$$S = \overline{A}.B + A.\overline{B} = A \oplus B$$

Associated Logigram



III.2 THE FULL ADDER

In binary, when you make an addition, you have to take into account the incoming carry.

$$\begin{array}{cccccc} & r_4 & r_3 & r_2 & r_1 & r_0 = 0 \\ & & a_4 & a_3 & a_2 & a_1 \\ + & & b_4 & b_3 & b_2 & b_1 \\ \hline & r_4 & s_4 & s_3 & s_2 & s_1 \end{array} \qquad \begin{array}{ccc} & & r_{i-1} \\ & & a_i \\ + & & b_i \\ \hline & r_i & s_i \end{array}$$

The one-bit full adder has 3 inputs:

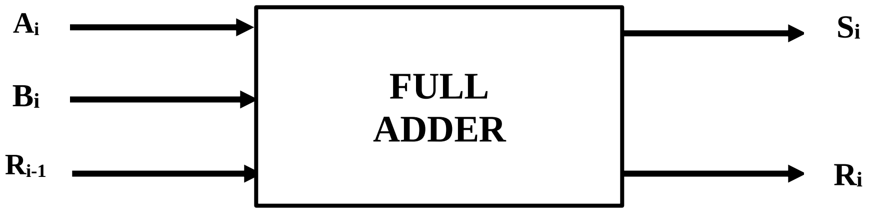
A_i : the first number on a bit.

B_i : the second number on one bit.

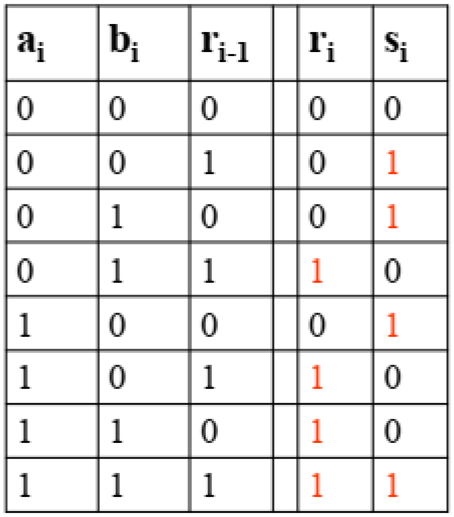
R_{i-1} : the incoming one-bit carry.

It has two outputs:
S: the sum

R_i : The outgoing Carry



The truth table of one-bit Full Adder :



a_i	b_i	r_{i-1}		r_i	s_i
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

$$S_i = \overline{A}_i.\overline{B}_i.R_{i-1} + \overline{A}_i.B_i.\overline{R}_{i-1} + A_i.\overline{B}_i.\overline{R}_{i-1} + A_i.B_i.R_{i-1}$$

$$R_i = \overline{A}_i B_i R_{i-1} + A_i \overline{B}_i R_{i-1} + A_i B_i \overline{R}_{i-1} + A_i B_i R_{i-1}$$

If we want to simplify the two logical formulas for Si and Ri, we get

$$S_i = \overline{A}_i.\overline{B}_i.R_{i-1} + \overline{A}_i.B_i.\overline{R}_{i-1} + A_i.\overline{B}_i.\overline{R}_{i-1} + A_i.B_i.R_{i-1}$$

$$S_i = \overline{A}_i.(\overline{B}_i.R_{i-1} + B_i.\overline{R}_{i-1}) + A_i.(\overline{B}_i.\overline{R}_{i-1} + B_i.R_{i-1})$$

$$S_i = \overline{A}_i(B_i \oplus R_{i-1}) + A_i.(\overline{B_i \oplus R_{i-1}})$$

$$S_i = A_i \oplus B_i \oplus R_{i-1}$$

$$R_i = \overline{A}_i B_i R_{i-1} + A_i \overline{B}_i R_{i-1} + A_i B_i \overline{R}_{i-1} + A_i B_i R_{i-1}$$

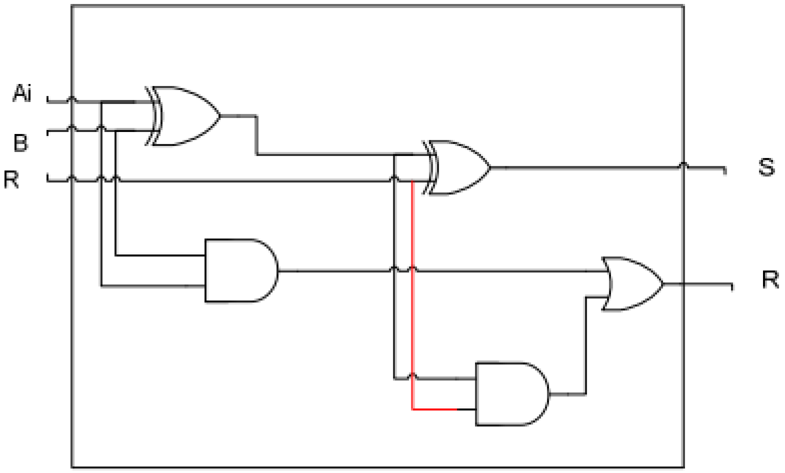
$$R_i = R_{i-1}.(\overline{A}_i.B_i + A_i.\overline{B}_i) + A_i B_i (\overline{R}_{i-1} +_i R_{i-1})$$

$$R_i = R_{i-1}.(A_i \oplus B_i) + A_i B_i$$

The one-bit Full Adder Logigram

$$R_i = A_i.B_i + R_{i-1}.(B_i \oplus A_i)$$

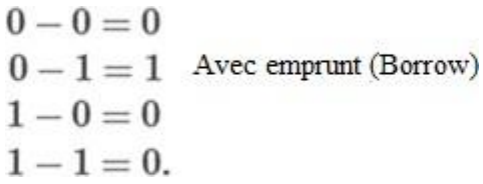
$$S_i = A_i \oplus B_i \oplus R_{i-1}$$



III.3 THE HALF SUBTRACTOR

The half-subtractor is a combinatorial circuit which performs the arithmetic subtraction of two numbers A and B on one bit. The result is the difference D and the borrow Br.

In binary, single-bit subtraction is performed as follows:



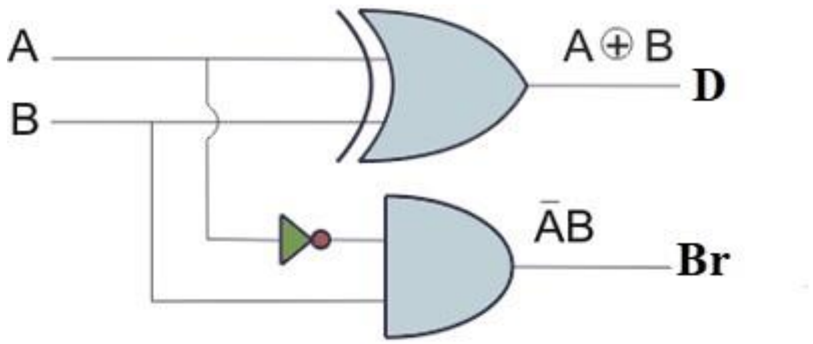
$$0 - 0 = 0$$
$$0 - 1 = 1 \quad \text{Avec emprunt (Borrow)}$$
$$1 - 0 = 0$$
$$1 - 1 = 0.$$

The associated truth table

A	B	Différence (D)	borrow (Br)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = A \oplus B$ et $Br = \bar{A}B$

The Diagram of the Half Sustractor



III.4 THE FULL SUBSTACTOR

The one-bit full subtractor has 3 inputs:

A_i : The first number on a bit.

B_i : The second number on a bit

InBr: The Input Borrow

D: Difference

Br : The borrow

Truth table for a complete one-bit subtractor

Ai	Bi	Input borrow (InBr)	Difference (D)	Borrow (Br)
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

$D = \bar{A}\ \bar{B}\ \mathrm{InBr} + \bar{A}\ \mathrm{B}\ \overline{InBr} + \mathrm{A}\ \bar{B}\ \overline{InBr} + \mathrm{A\ B\ InBr}$

$D = \bar{A}\ (\bar{B}\ \mathrm{InBr} + \mathrm{B}\ \overline{InBr}) + \mathrm{A}\ (\bar{B}\ \overline{InBr} + \mathrm{B\ InBr})$

$D = \bar{A}\ (B \oplus \mathrm{InBr}\ ) + \mathrm{A}\ \overline{(B \oplus InBr\ )}$

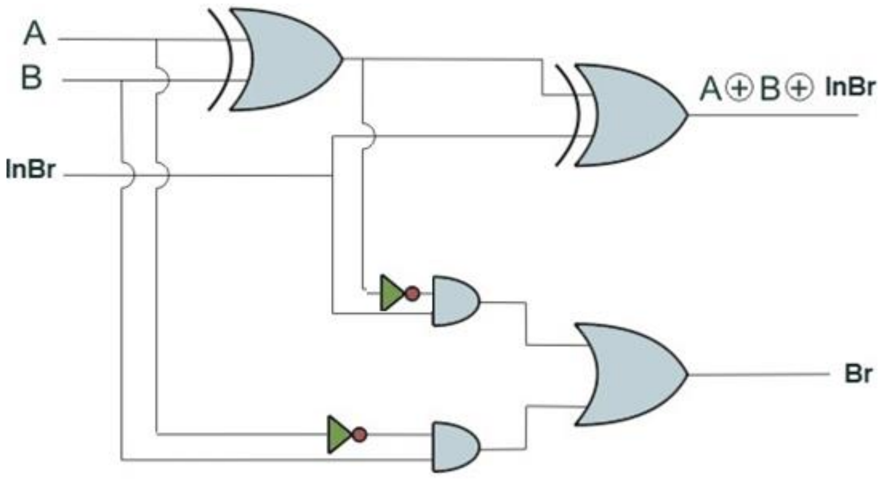
$D = A \oplus B \oplus InBr$

$Br = \bar{A}\ \bar{B}\ InBr + \bar{A}\ B\ \overline{InBr} + \bar{A}\ B\ InBr + \mathrm{A\ B\ InBr}$

$Br = (\ \bar{A}\ \bar{B} + \mathrm{AB}\ )\ \mathrm{InBr} + \bar{A}\ B\ (\ \overline{InBr} + \mathrm{InBr})$

$Br = (\ \overline{A \oplus \mathrm{B}})\ \mathrm{InBr} + \bar{A}\ B$

The Full Substractor Diagram



III.4 A ONE BIT COMPARATOR

A logic comparator is a logic circuit that compares two binary numbers, usually labelled A and B. It has 3 possible outputs noted :

A = B (A equals B)

A > B (A is strictly greater than the number B) and A < B (A is strictly less than the number B) Of which :

If A = B, the output A = B goes to state 1 while the outputs A > B and A < B go to state 0. If the number A is strictly greater than the number B, then the output A> B goes to state 1 while the outputs A = B and A < B go to state 0.

If the number A is strictly less than the number B, only the output A < B goes to state 1.

Truth table for a one-bit comparator

A	B		fs	fe	fi
0	0		0	1	0
0	1		0	0	1
1	0		1	0	0
1	1		0	1	0

$$fs = A.\overline{B}$$

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

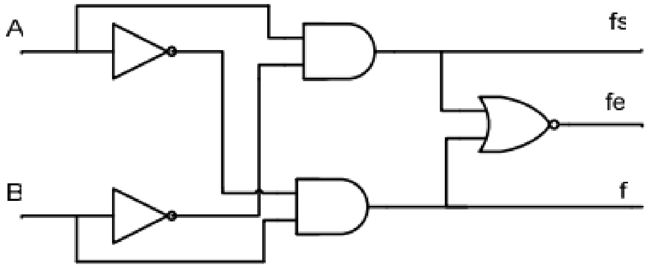
$$fe = \overline{A}\,\overline{B} + AB = \overline{A \oplus B} = \overline{fs + fi}$$

L

$$fs = A.\overline{B}$$

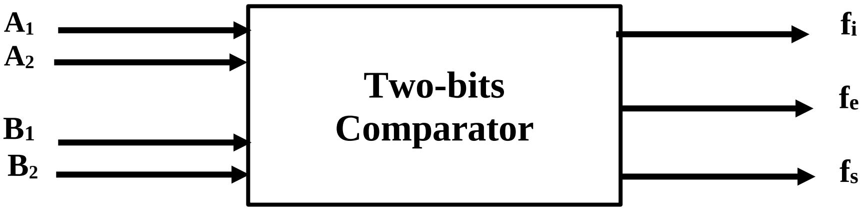
$$fi = \overline{A}B$$

$$fe = \overline{fs + fi}$$



III.5 THE TWO-BITS COMPARATOR :

It compares two numbers A (a2a1) and B (b2b1), each on two bits.



1. A=B si

A2=B2 et A1=B1

$$fe = (\overline{A2 \oplus B2}).(\overline{A1 \oplus B1})$$

2. A>B si

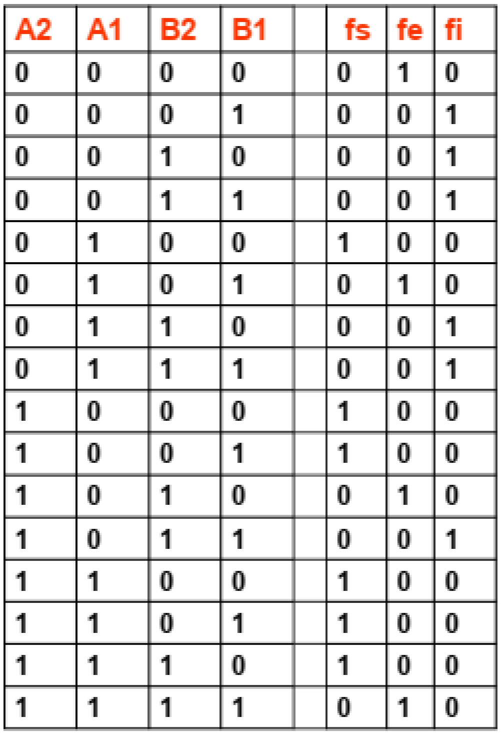
A2 > B2 ou (A2=B2 et A1>B1)

$$fs = A2.\overline{B2} + (\overline{A2 \oplus B2}).(A1.\overline{B1})$$

3. A<B si

A2 < B2 ou (A2=B2 et A1<B1)

$$fi = \overline{A2}.B2 + (\overline{A2 \oplus B2}).(\overline{A1}.B1)$$



A2	A1	B2	B1		fs	fe	fi
0	0	0	0		0	1	0
0	0	0	1		0	0	1
0	0	1	0		0	0	1
0	0	1	1		0	0	1
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	1
1	0	0	0		1	0	0
1	0	0	1		1	0	0
1	0	1	0		0	1	0
1	0	1	1		0	0	1
1	1	0	0		1	0	0
1	1	0	1		1	0	0
1	1	1	0		1	0	0
1	1	1	1		0	1	0

III.6 THE MULTIPLEXER

A multiplexer is a logic circuit with several data inputs, but only one output which communicates the data. The routing of input data to the output is controlled by the Select inputs (or address input).

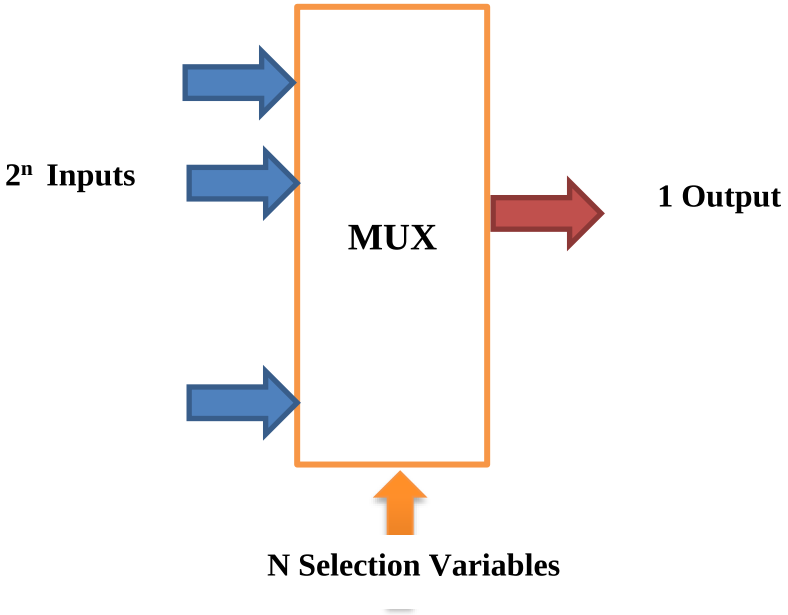
A multiplexer is a combinatorial circuit which can select one item of information (1 bit) from 2n input values.

It has :

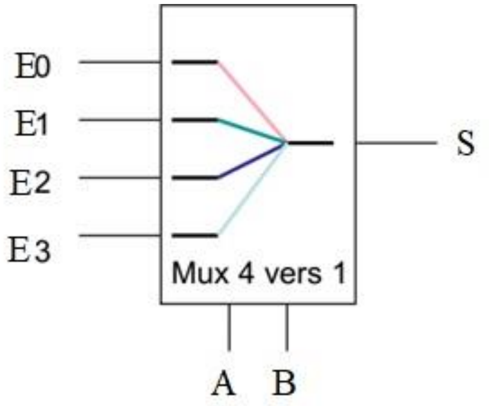
2n information inputs

A single output

N selection inputs (commands)



Multiplxer 4→1

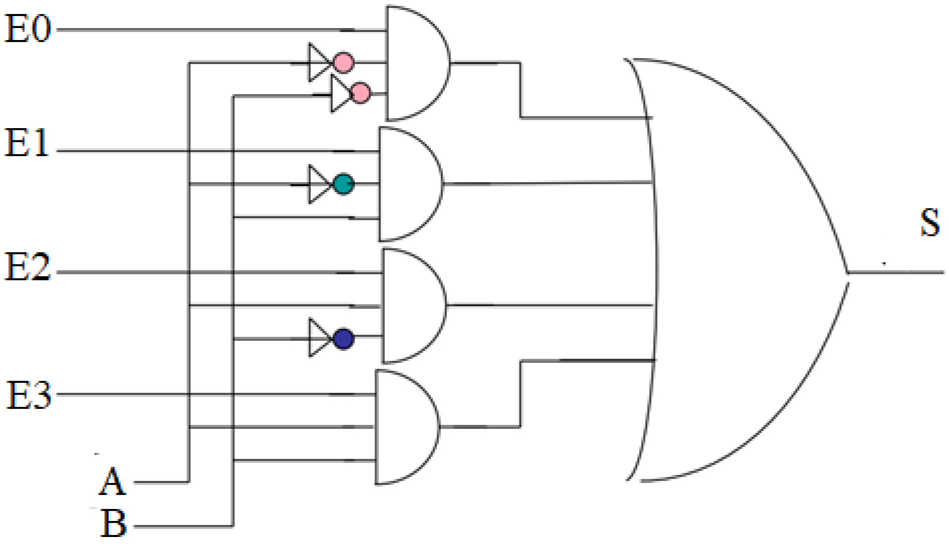


The Truth table

A	B	S
0	0	E_0
0	1	E_1
1	0	E_2
1	1	E_3

$S = \overline{A} \,.\, \overline{B} \,.\, E_0 + \overline{A} \,.\, B.\, E_1 + A \,.\, \overline{B} \,.\, E_2 + A.B.E_3$

The Associate Diagram



In this case, there are 4 inputs E_0, E_1, E_2, E_3 which are transmitted to the output according to the choice indicated

by one of the four possible combinations of selection outputs A and B.

III.7 THE DEMULTIPLEXER

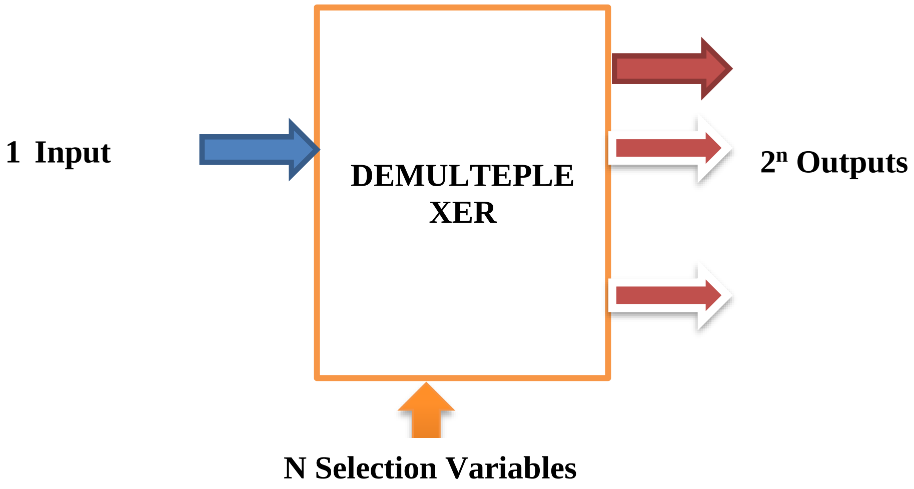
A demultiplexer is a logic circuit with a single input and N output channels. It receives

the input data and chooses to direct it to one of the N possible output channels,

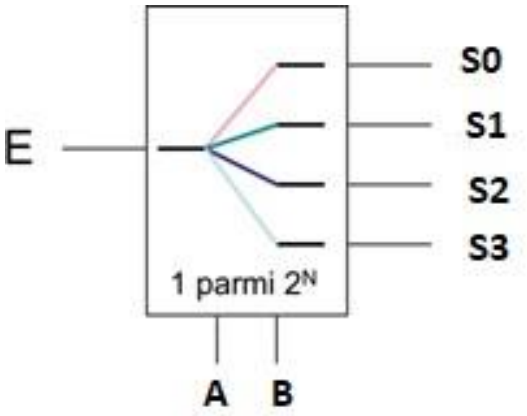
It plays the opposite role to a multiplexer, allowing information to be passed to one of the outputs according to the values of the control inputs

It has :

- A single input
- 2n outputs
- N selection inputs



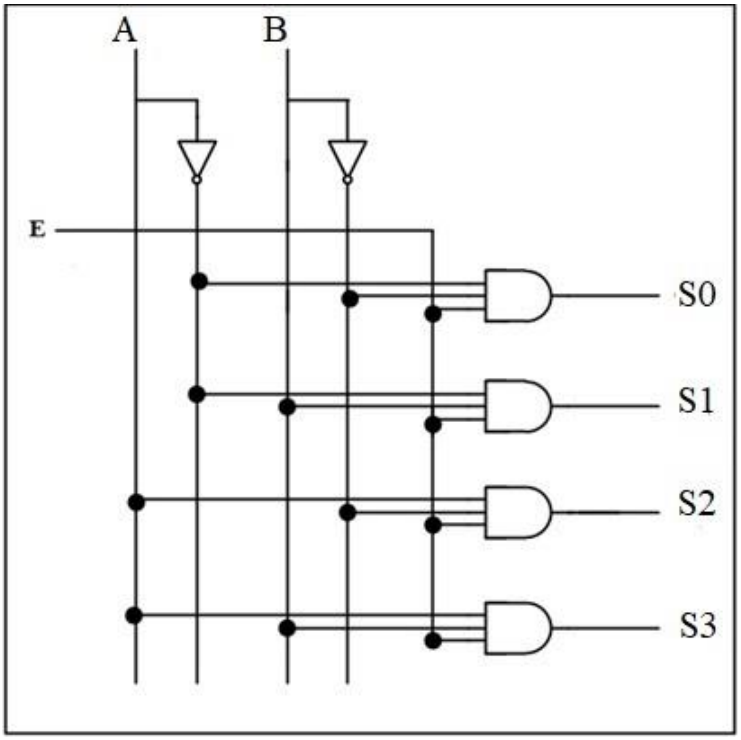
Demultiplexer 4→1



Associate Truth Table :

A	B	S0	S1	S2	S3
0	0	**E**	0	0	0
0	1	0	**E**	0	0
1	0	0	0	**E**	0
1	1	0	0	0	**E**

Associate Diagram



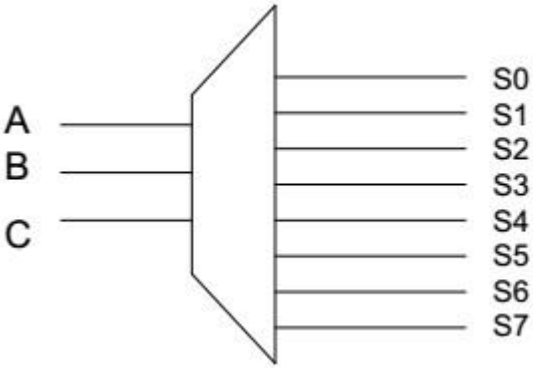
$$S0 = \overline{A}.\overline{B}\,(E)$$
$$S1 = \overline{A}.B\,.(E)$$
$$S2 = A.\overline{B}\,.(E)$$
$$S3 = A.B\,(E)$$

III.8 THE BINARY DECODER :

It is a combinatorial circuit consisting of :

- N: data inputs
- 2n outputs

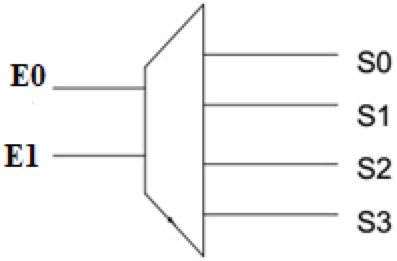
For each input combination, only one output is active at a time.



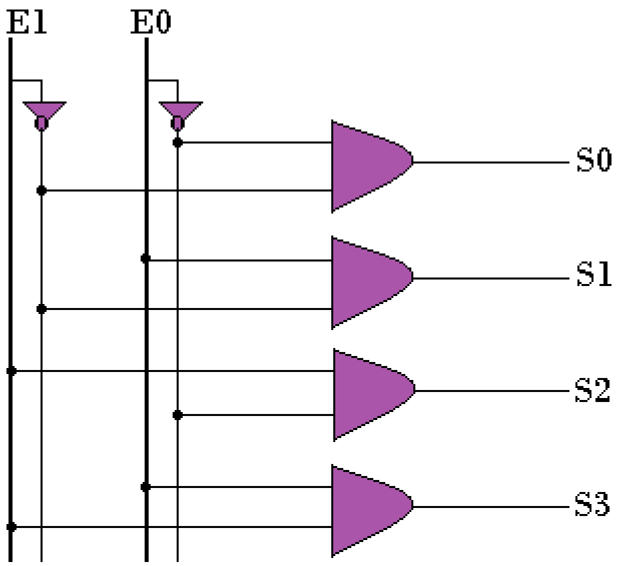
A DECODER 3→8

THE TRUTH TABLE OF A DECODER 2→4

E_1	E_2	S0	S1	S2	S3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



$S_0 = \overline{E1}.\overline{E0}$, $S_1 = \overline{E1}.E0$, $S_2 = E1.\overline{E0}$, $S_3 = E1\ E0$

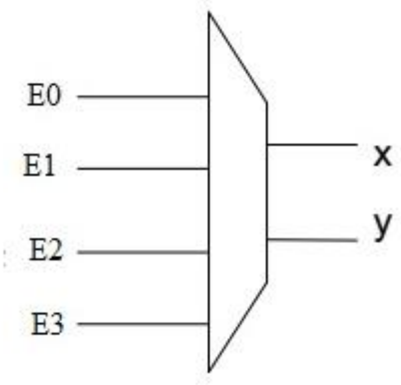


III.9 THE BINARY ENCODER

It plays the opposite role to a decoder, consisting of :

- **2^n inputs**
- **N outputs**

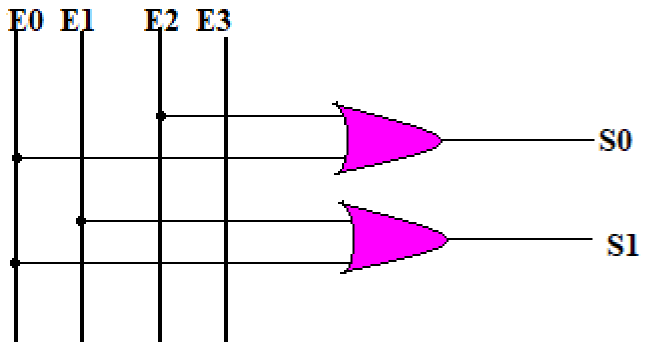
For each input combination we will have its number (in binary) at the output



ENCODER 4→2

E3	E2	E1	E0	S_1	S_0
0	0	0	0	0	0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

S1 = E1 + E0   S0 = E2 + E0



III.10 THE PRIORITY ENCODER

Often in an encoder, then called a priority encoder, if several inputs are active simultaneously, in this case we must choose which input bit priority we are going to choose in what follows is a high-order priority encoder.

Truth table

E3	E2	E1	E0	S1	S0
0	0	0	0	**0**	**0**
0	0	0	1	**0**	**0**
0	0	1	0	**0**	**1**
0	0	1	1	**0**	**1**
0	1	0	0	**1**	**0**
0	1	0	1	**1**	**0**
0	1	1	0	**1**	**0**
0	1	1	1	**1**	**0**
1	0	0	0	**1**	**1**
1	0	0	1	**1**	**1**
1	0	1	0	**1**	**1**
1	0	1	1	**1**	**1**
1	1	0	0	**1**	**1**
1	1	0	1	**1**	**1**
1	1	1	0	**1**	**1**
1	1	1	1	**1**	**1**

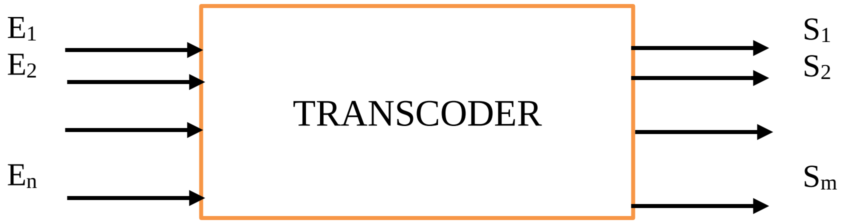
After simplification (Karnaugh table)

$S_1 = E2 + E3$

$S_0 = E3 + \overline{E2}\,E1$

III.11 THE TRANSCODER :

- A transcoder is a combinatorial circuit that transforms an input X code (n bits) into an output Y code (m bits).



A transcoder transforms information available at the input in a given form (generally a code) into the same information, but in another form (generally a different code)

The two most important applications for transcoders are: code conversion and segment display.