Combinational Logic circuit

Chapter:- 6

Combinational Circuit

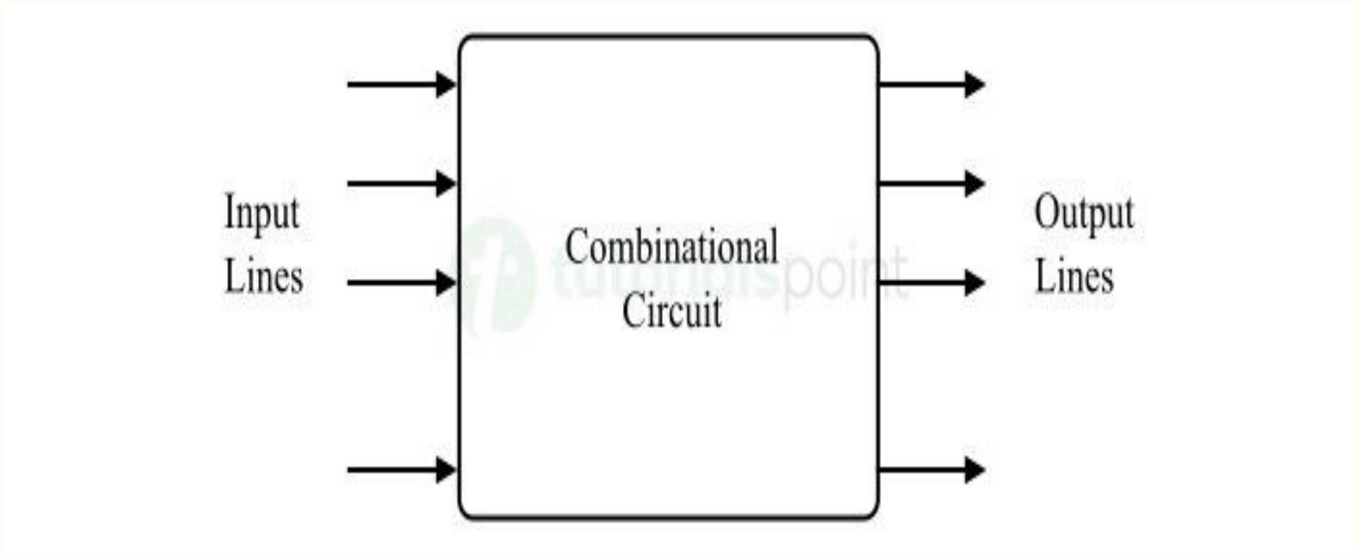
A combinational circuit is a type of digital logic circuit whose output depends on the present input values only and does not depend on past input and output values. Therefore, a combinational circuit is considered to not have a memory element in its circuit that stores previous inputs and outputs.

The most important characteristic of a combinational circuit is that it does not have any feedback path between input and output. Therefore, the combinational circuits can be categorized as open-loop systems.

Combinational Circuit

Block Diagram of Combinational Circuit

The following figure depicts the block diagram of a combinational logic circuit.



Here, we can see that there are only three key elements in the circuit diagram of a combinational circuit, they are –

Combinational Circuit

Input Lines − The input lines are used to enter the input values into the combinational circuit.
Processing Unit − It is the main element that processes the input values depending on the type of the circuit. For example, a full adder adds three binary bits.
Output Lines − The output lines are used to take results generated by the circuit.

Characteristics of Combinational Circuit

The following are the main characteristics of combinational circuits −

- The output of a combinational circuit, at any instant of time, depends only on the present input values at that instant of time.
- Combinational circuits do not use any kind of memory element in their circuits. Thus, the previous state of input and output values do not have any effect on the present operation of the circuit.
- The output of a combinational circuit can be entirely predicted using its logical operation and input values.
- Combinational circuits produce an instantaneous output in response to any change in its input values.

Types of Combinational Circuit

The combinational circuits are important components of digital systems. Depending on the functions performed, there are various types of combinational circuits. Some common types of combinational circuits and their functions are explained below −

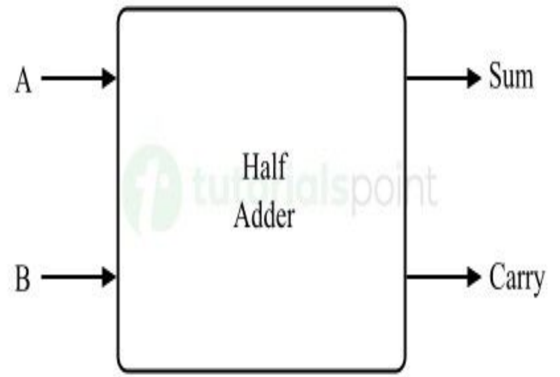
- Binary Adders
- Binary Subtractors
- Multiplexers (MUX)
- Demultiplexers (DEMUX)
- Encoders
- Decoders

Binary Adders

A **binary adder** is a combinational circuit that performs the addition of binary digits or bits. Depending on the design and configuration, there are two types of binary adders namely, Half Adder and Full Adder.

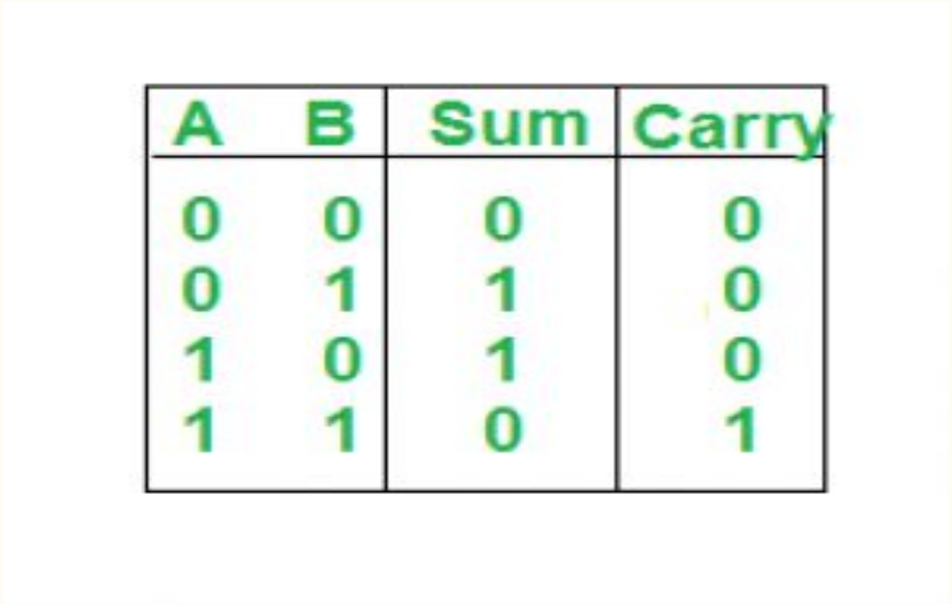
Half Adder

The **half adder** is a combinational logic circuit with two inputs and two outputs. The half adder circuit is designed to add two single-bit binary numbers A and B. It is the basic building block for the addition of two single-bit numbers. This circuit has two outputs namely, sum and carry.



Half Adders

Truth Table half adder:

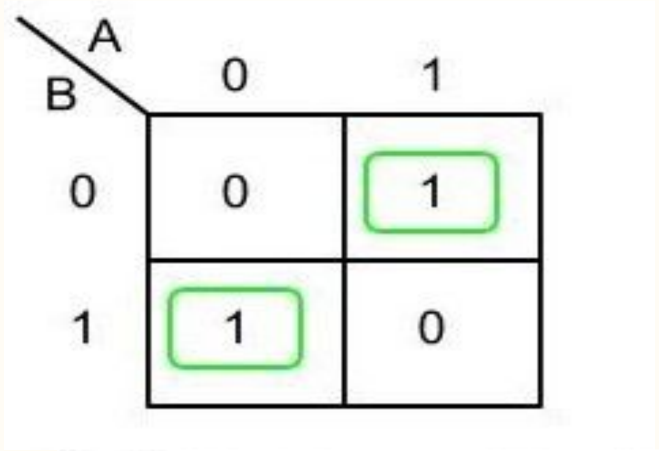


A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half Adders

Logical expression of half adder using k-map:

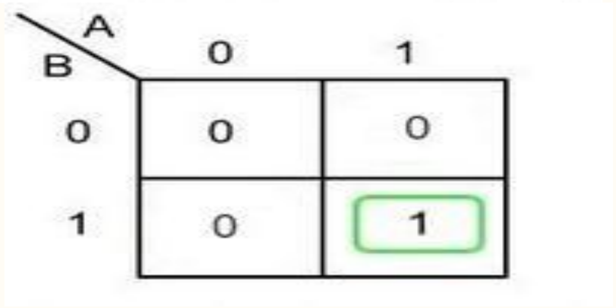
For sum



B \ A	0	1
0	0	1
1	1	0

Sum = A XOR B

For Carry

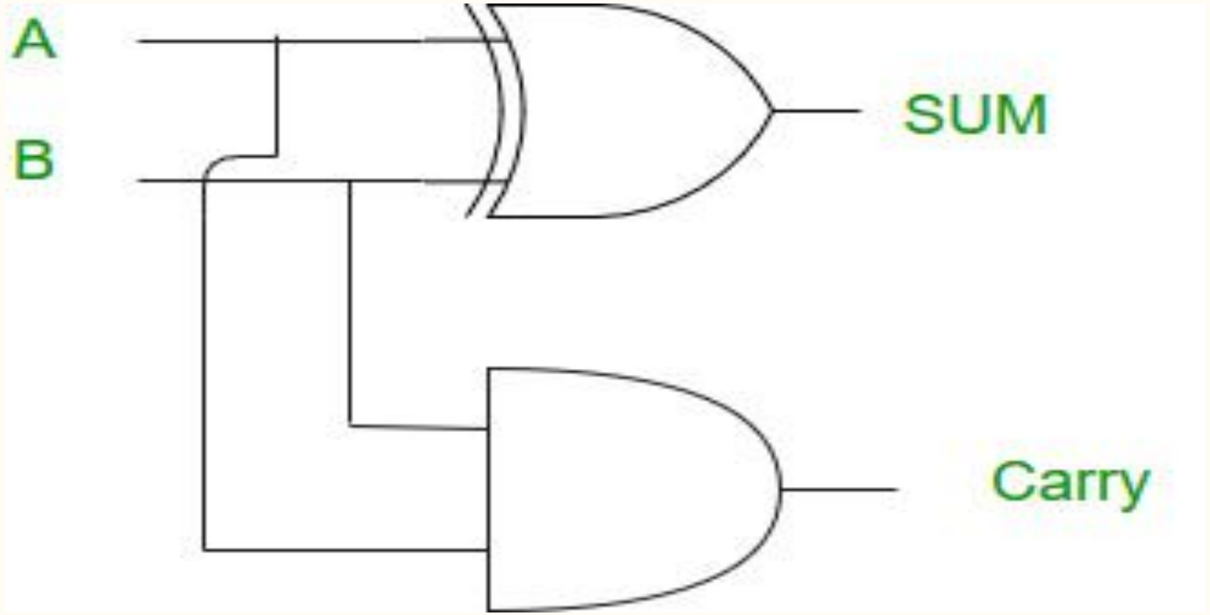


B \ A	0	1
0	0	0
1	0	1

Carry = A and B

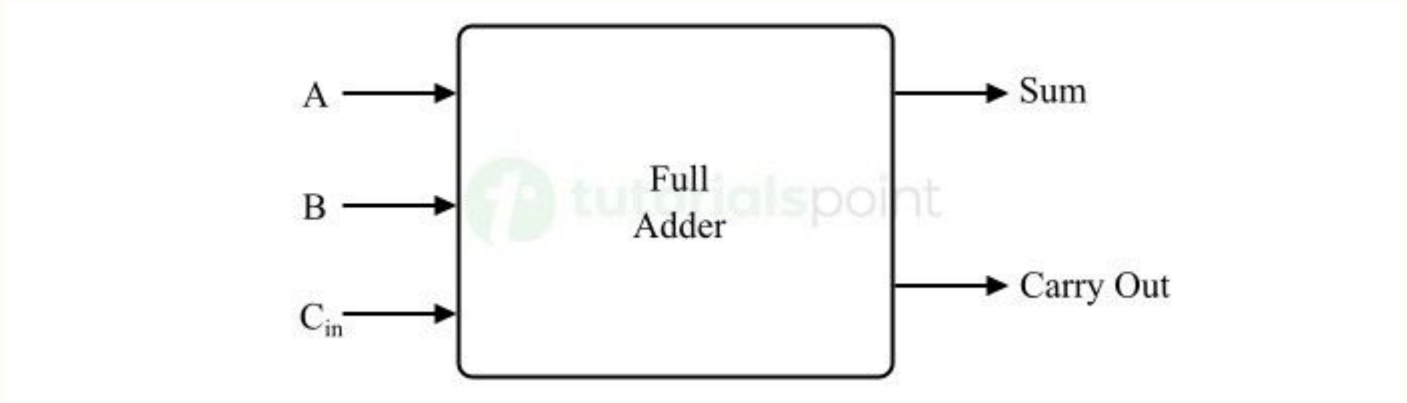
Half Adders

Logical expression of half adder implementation:



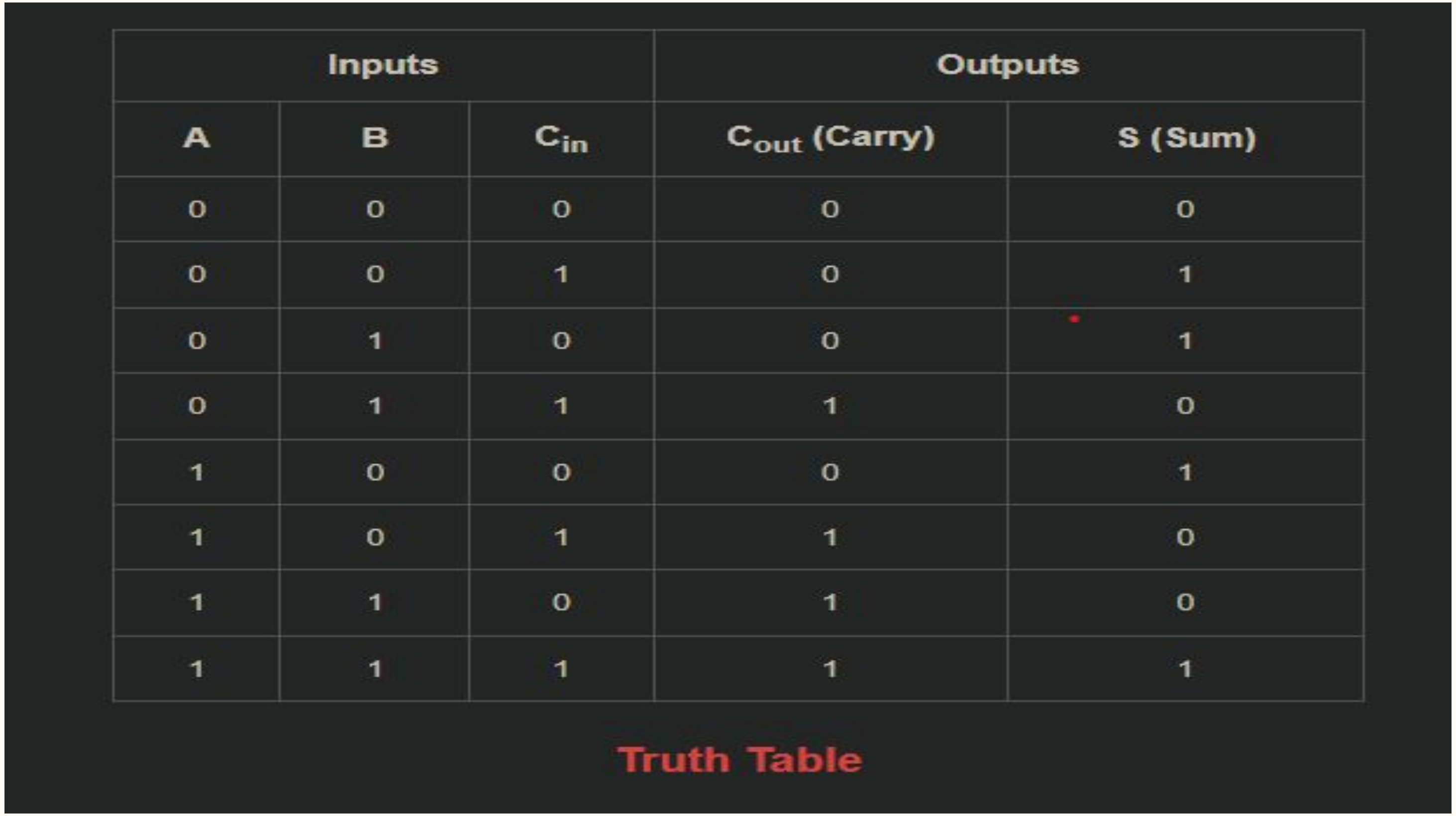
Full Adders

The **full adder** is designed to overcome the drawback of a half adder which is the ability to add only two bits. Therefore, the full adder is a three-input and two-output combinational circuit. Where, the inputs are two one-bit numbers A and B, and a carry C from the previous addition. The outputs are sum and carry output.



Full Adders

Truth Table Full adder:

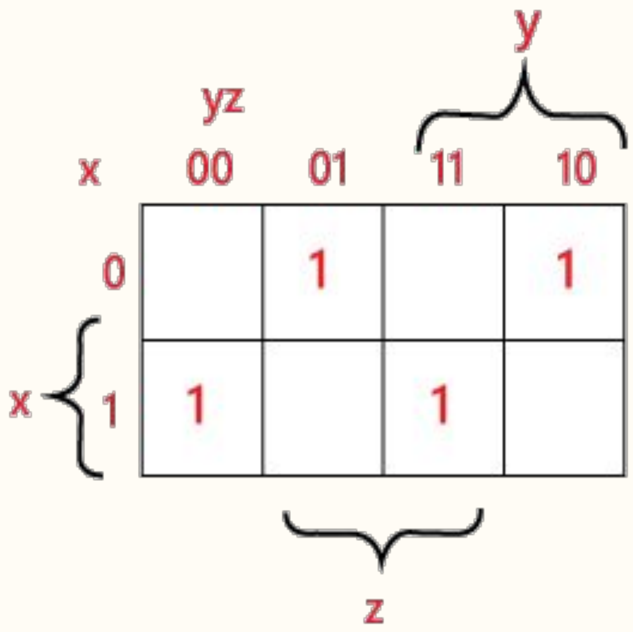


Inputs			Outputs	
A	B	C_{in}	C_{out} (Carry)	S (Sum)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

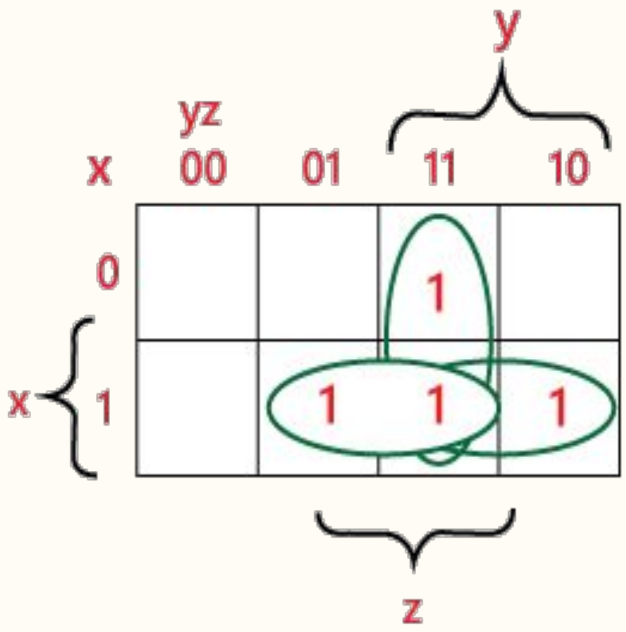
Truth Table

Full Adders

K-map of Full adder:



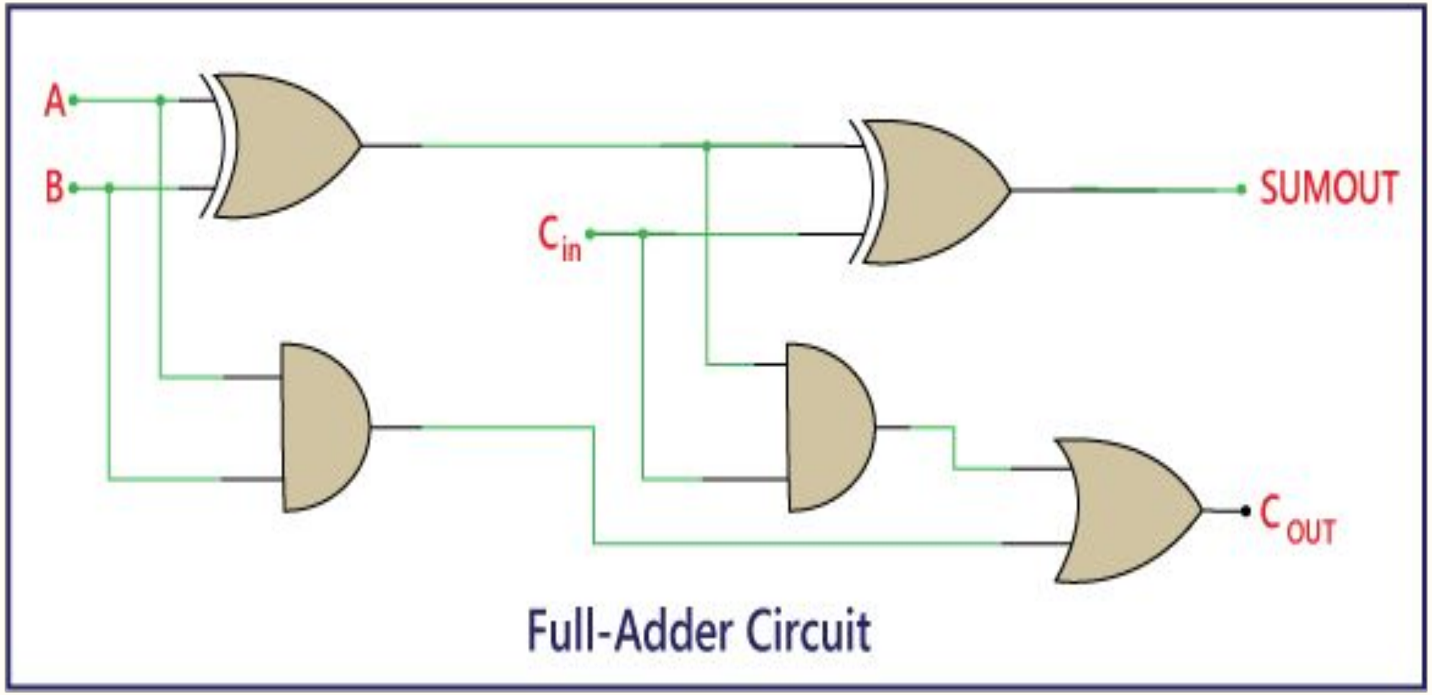
$S = x'y'z + x'yz' + xy'z' + xyz$



$C = xy + xz + yz$

Full Adders

Logical expression of Full adder implementation:

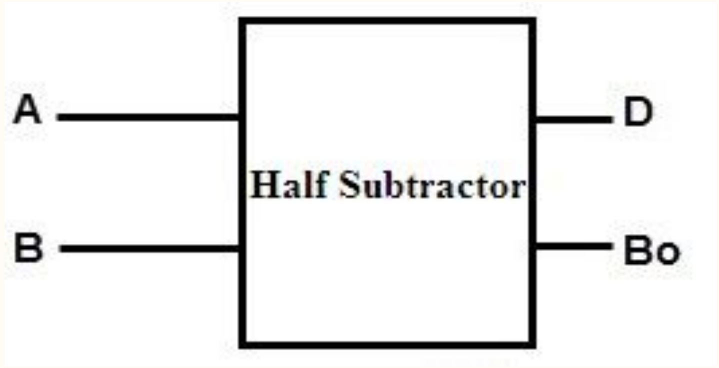


Subtractors

Half Subtractor is a kind of ‘Combinational Circuit’. These are designed to achieve the difference between two given binary inputs. The numbers used for this process are specified as ‘Minuend’ and ‘Subtrahend’. The value of ‘subtrahend’ will be subtracted from the ‘Minuend’ during this process. For the applied two inputs two output bits are generated.

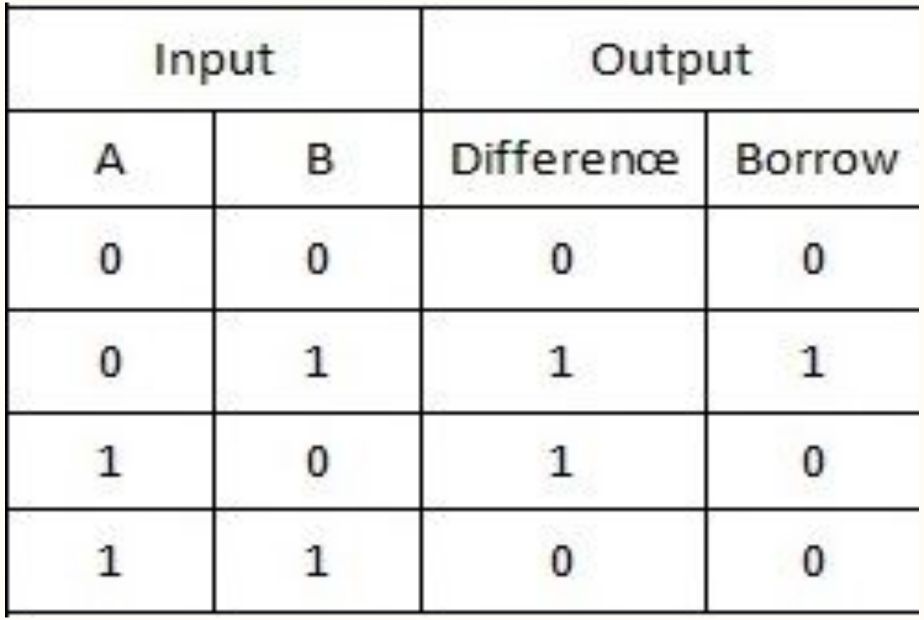
Half Subtractor

The ‘Combinational Circuit’ of this Half Subtractor consists of the inputs ‘A and B’. The outputs generated can be indicated as Difference (D) and the Borrow (Bo). The borrow bit is to indicate whether the bit has been taken from the previous stage.



Half Subtractor

Truth Table half subtractors:

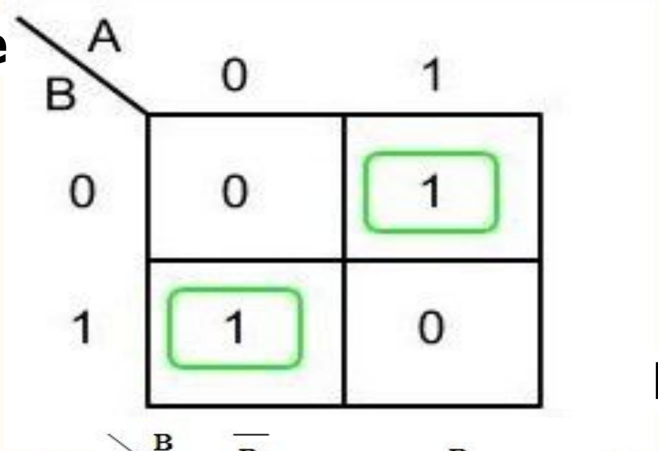


Input		Output	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Half Subtractor

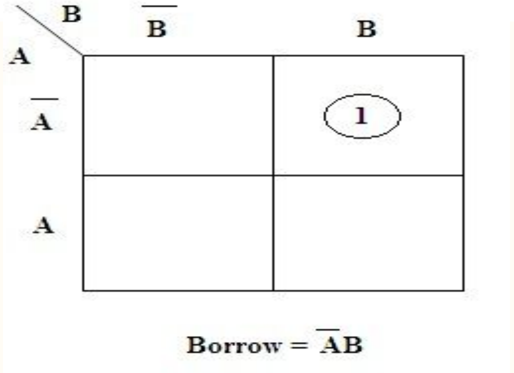
Logical expression of half subtractor using k-map:

For difference



Difference= A XOR B

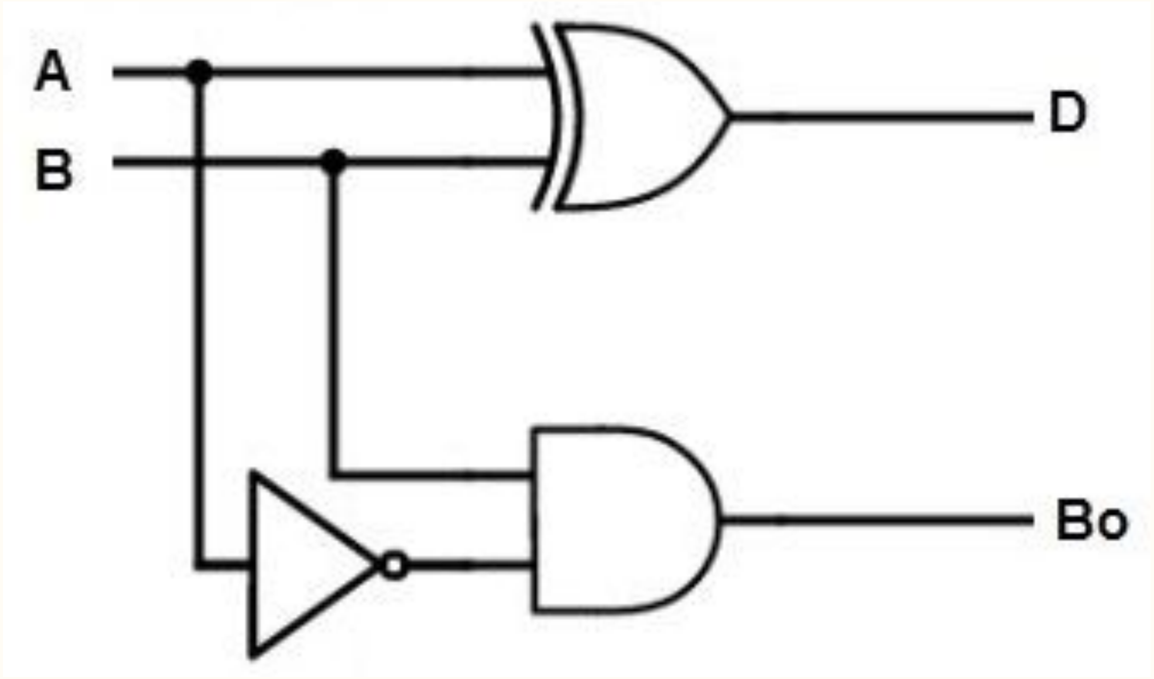
For borrow



Borrow= A′ and B

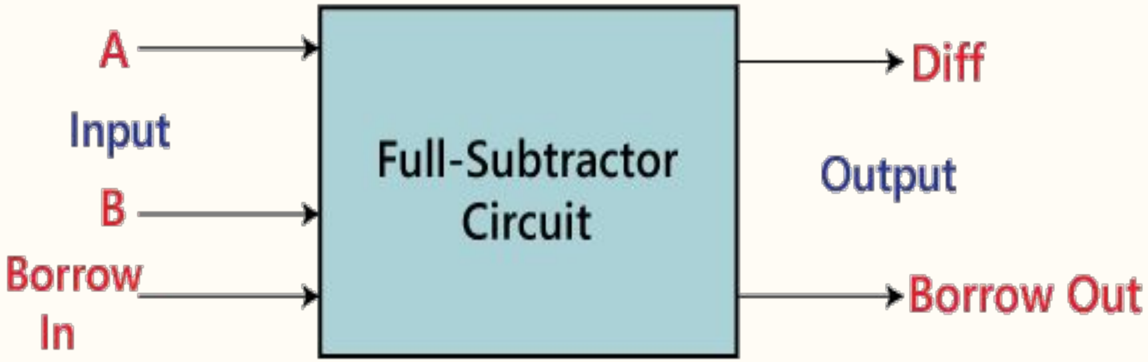
Half Subtractor

Logical expression of half subtractor implementation:



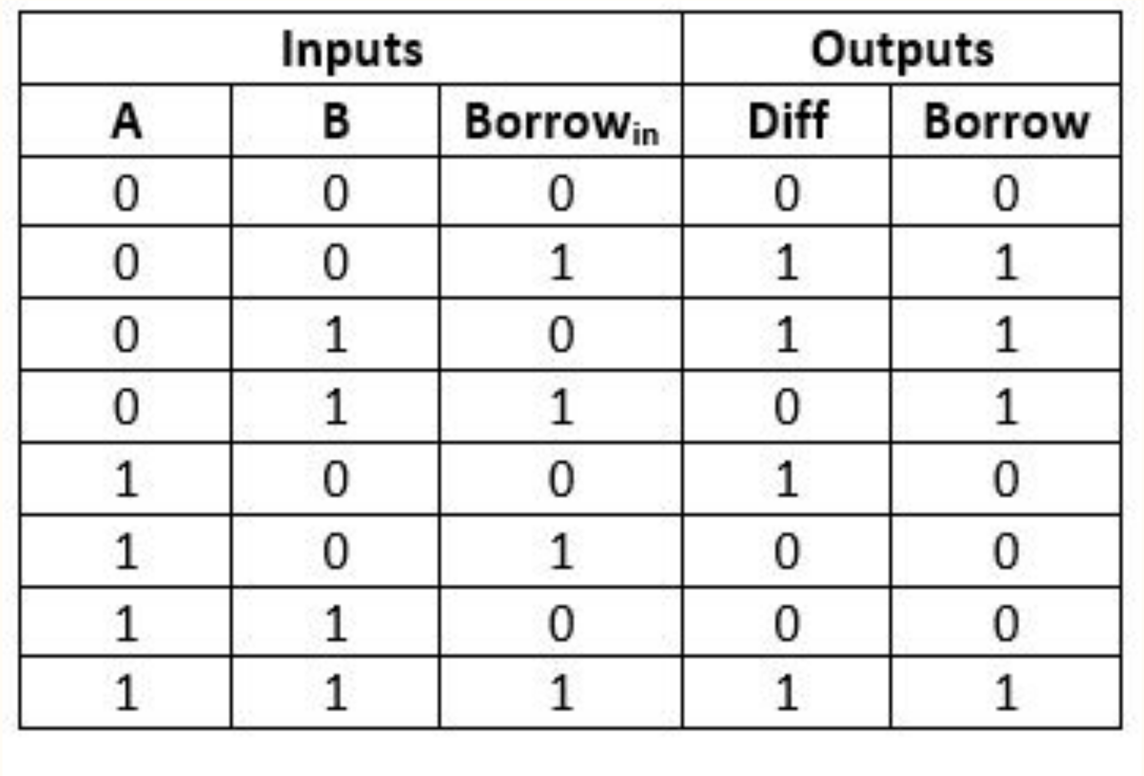
Full Subtractors

The Half Subtractor is used to subtract only two numbers. To overcome this problem, a full subtractor was designed. The full subtractor is used to subtract three 1-bit numbers A, B, and C, which are minuend, subtrahend, and borrow, respectively. The full subtractor has three input states and two output states i.e., diff and borrow.



Full Subtractor

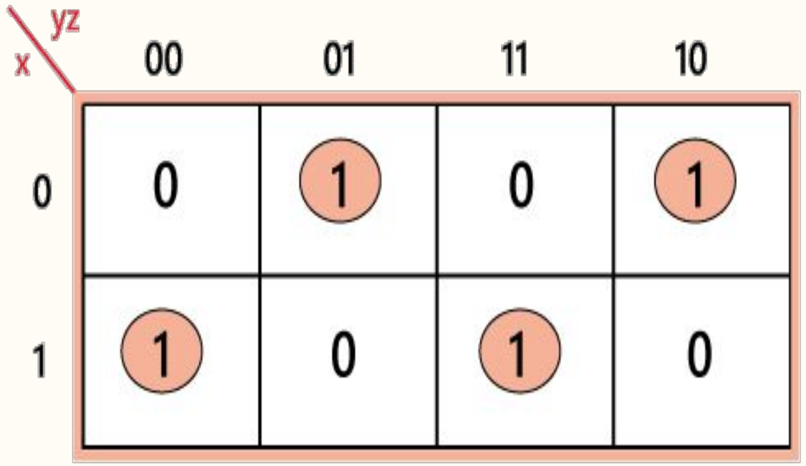
Truth Table Full subtractor:



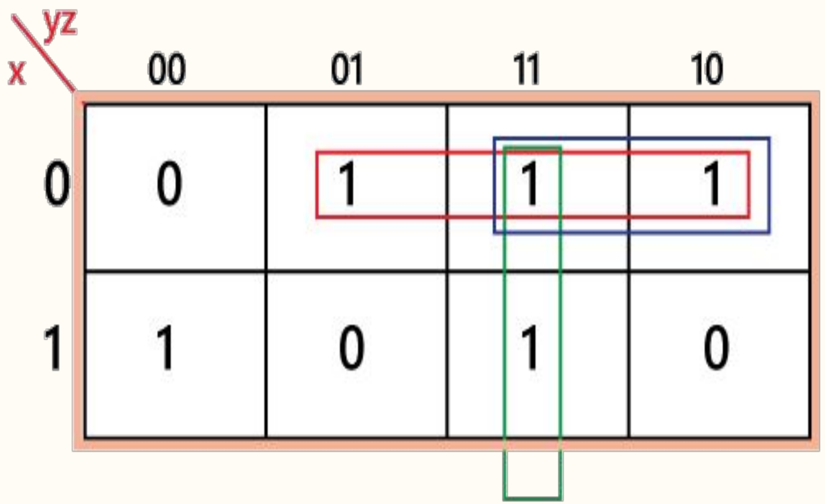
Inputs			Outputs	
A	B	Borrow_{in}	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Full Subtractor

K-map of Full subtractor:



x \ yz	00	01	11	10
0	0	1	0	1
1	1	0	1	0



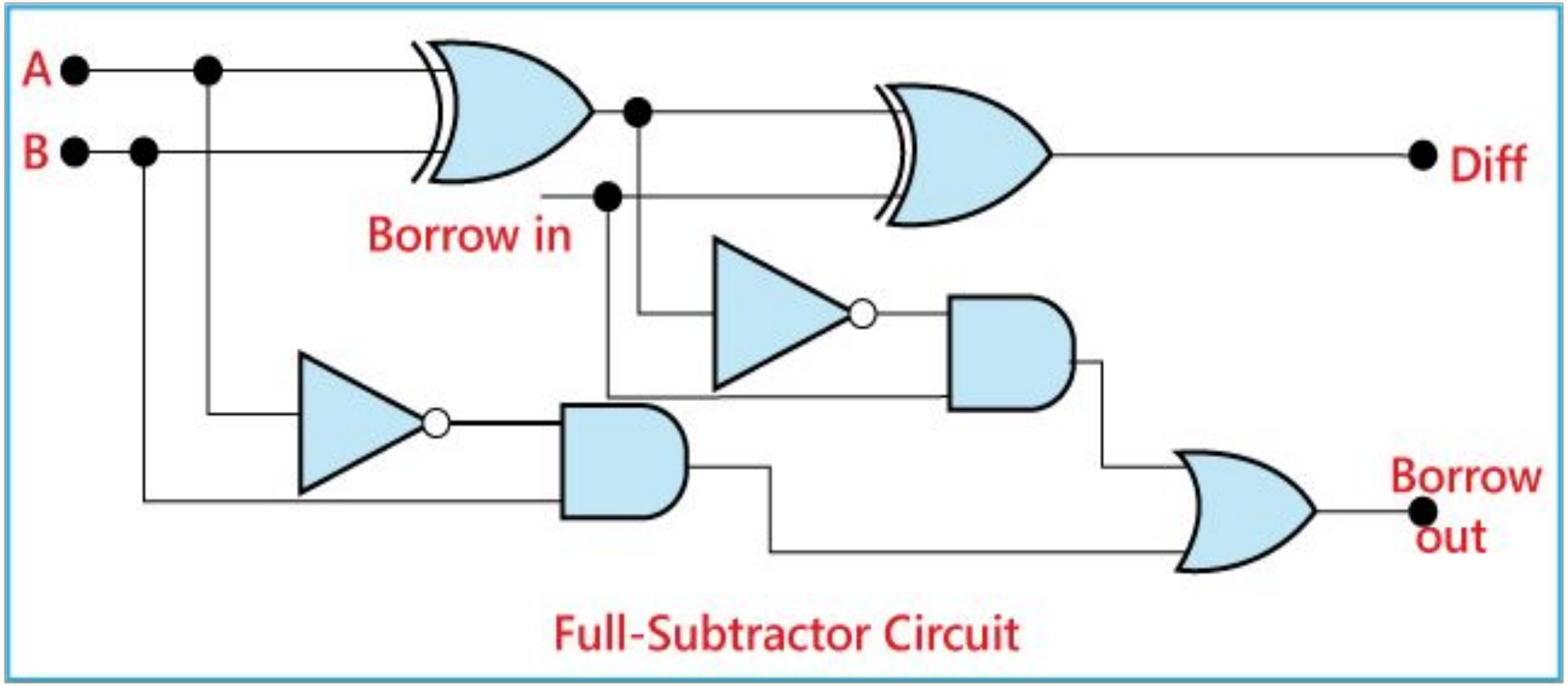
x \ yz	00	01	11	10
0	0	1	1	1
1	1	0	1	0

Diff = xy′z′ + x′yz′ + x′y′z + xyz

Borrow = x′y + x′z + yz

Full Subtractor

Logical expression of Full Subtractor implementation:



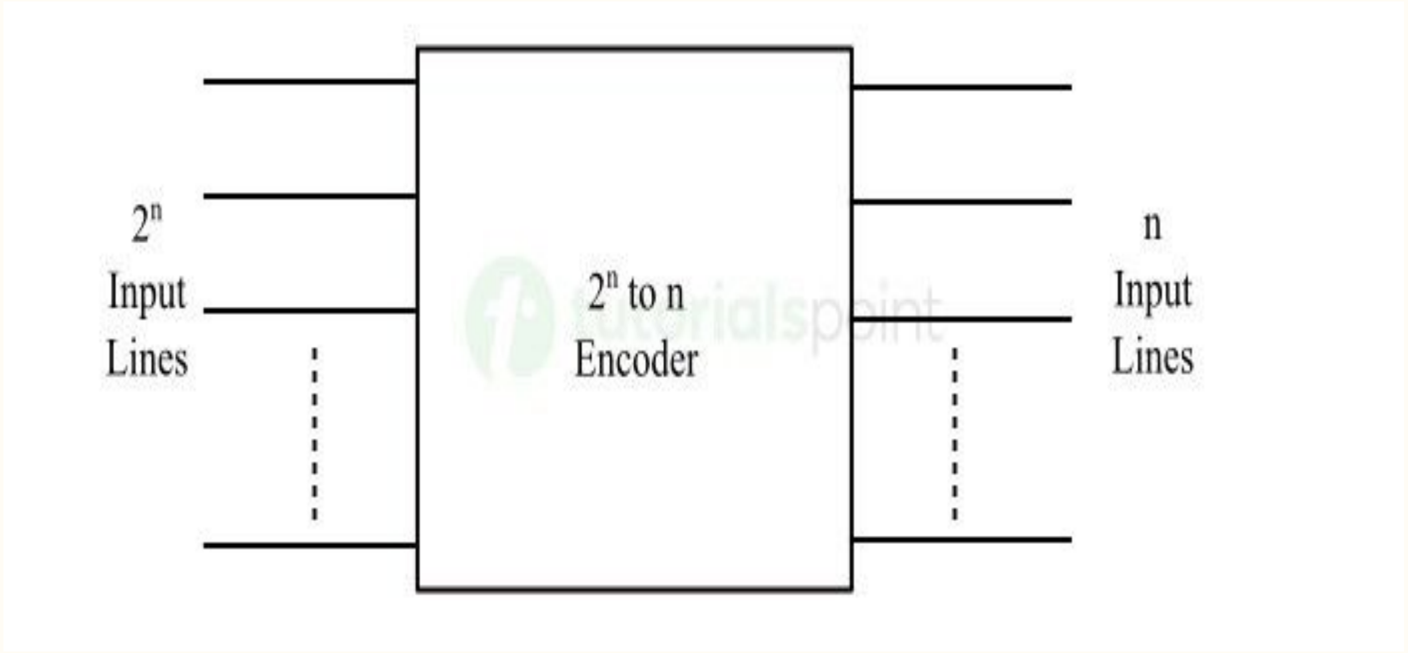
Encoder

An **encoder** is a digital combinational circuit that converts a human friendly information into a coded format for processing using machines. In simple words, an encoder converts a piece of information normal form to coded form. This process is called encoding.

An encoder consists of a certain number of input and output lines. Where, an encoder can have maximum of "2^n" input lines whereas "n" output lines. Hence, an encoder encodes information represented by "2^n" input lines with "n" bits.

Encoder

The **block diagram of an encoder** is shown in the following figure −



Encoder

Some of the commonly used types of encoders in digital electronics −

- 4 to 2 Encoder
- 8 to 3 Encoder (Octal Encoder)
- Decimal to BCD Encoder

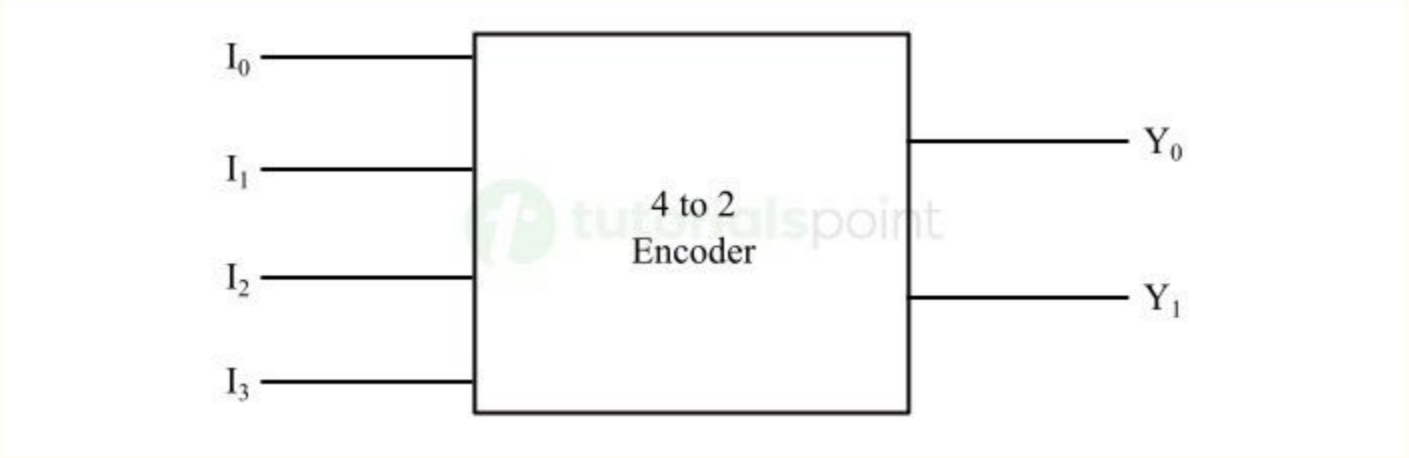
Let us now discuss these three types of most commonly used encoders in detail.

Encoder

4 to 2 Encoder

A 4 to 2 Encoder is a type of encoder which has 4 (2^2) input lines and 2 output lines. It produces an output code (i.e., convert input information in a 2-bit format) depending on the combination of input lines.

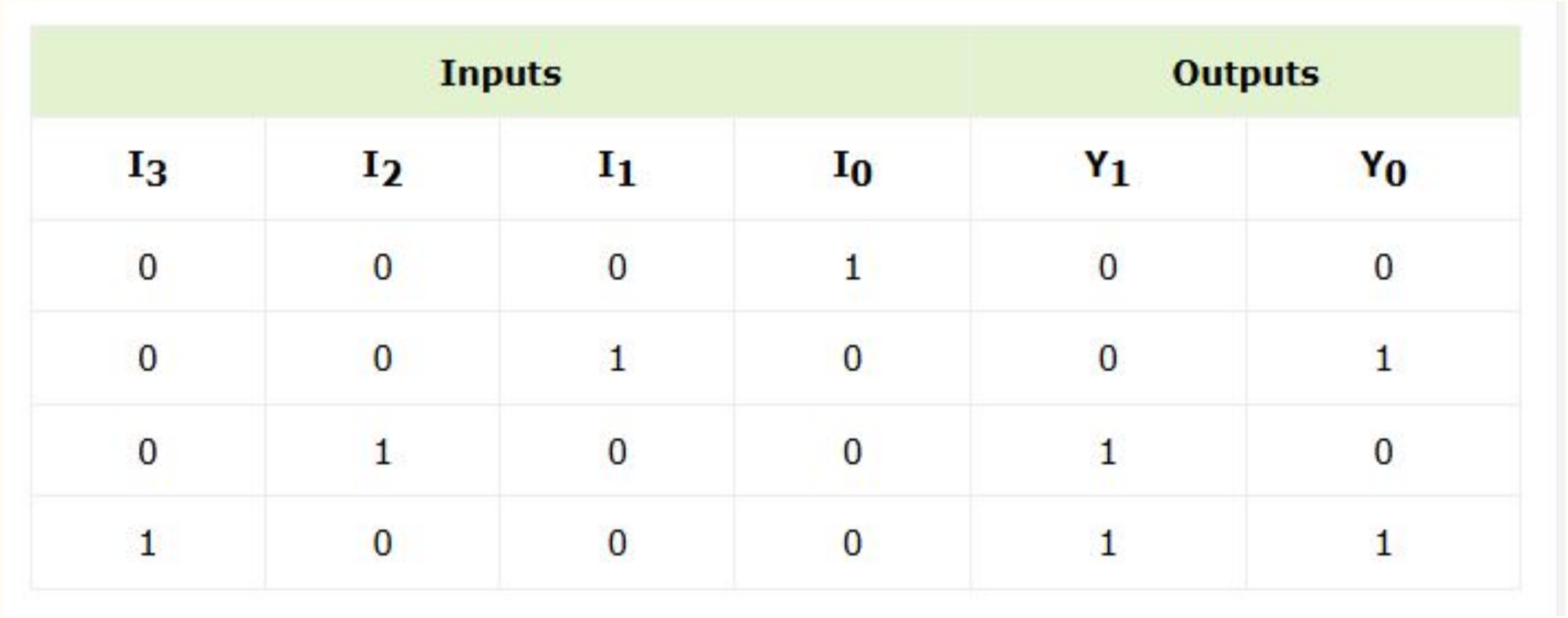
The block diagram of a 4 to 2 Encoder is shown in the following figure.



Encoder

4 to 2 Encoder

The working of a 4 to 2 Encoder for different input combinations is described in the following truth table



Inputs				Outputs	
I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Encoder

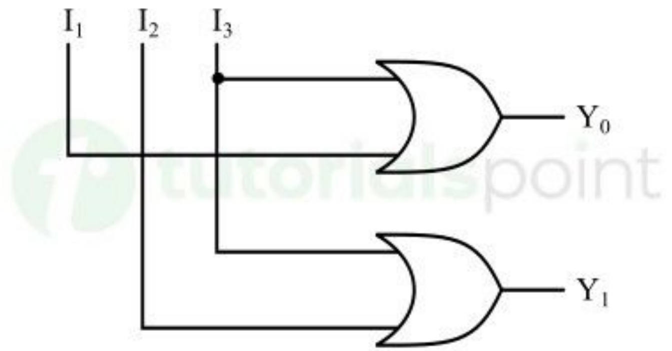
4 to 2 Encoder

From this truth table, we can derive the Boolean expression for each output of the 4 to 2 Encoder as follows −

$$Y_0 = I_1 + I_3$$

$$Y_1 = I_2 + I_3$$

It is clear that we can implement the logic circuit of the 4 to 2 Encoder using two OR gates. The following figure depicts the logic diagram of the 4 to 2 Encoder.

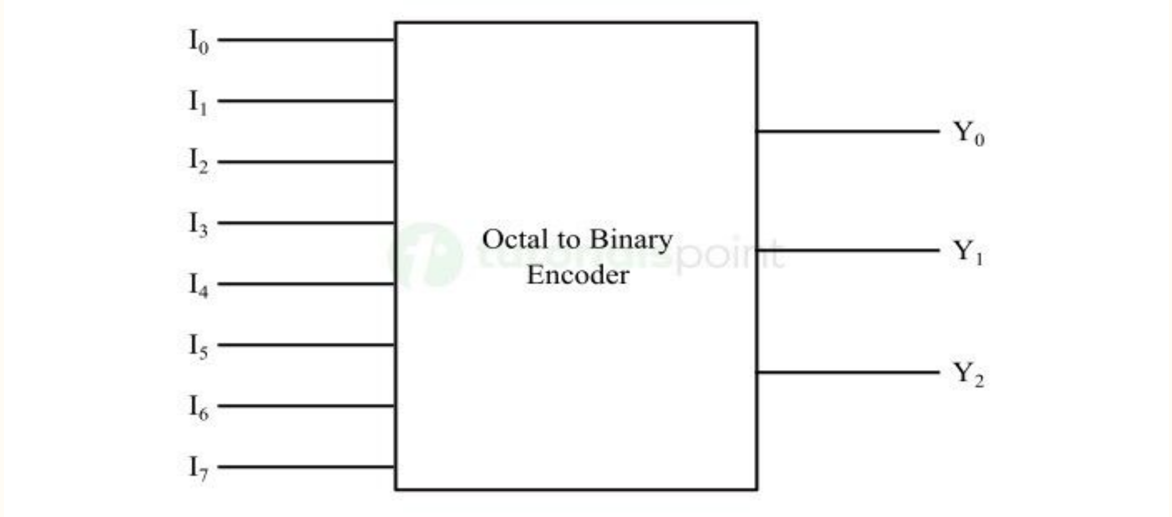


Encoder

Octal to Binary Encoder

The octal to binary encoder is a type of encoder that converts an octal code into binary code. It accepts 8 input lines and produces a 3-bit output depending on the combination of input lines. Therefore, it is also known as **8 to 3 Encoder**.

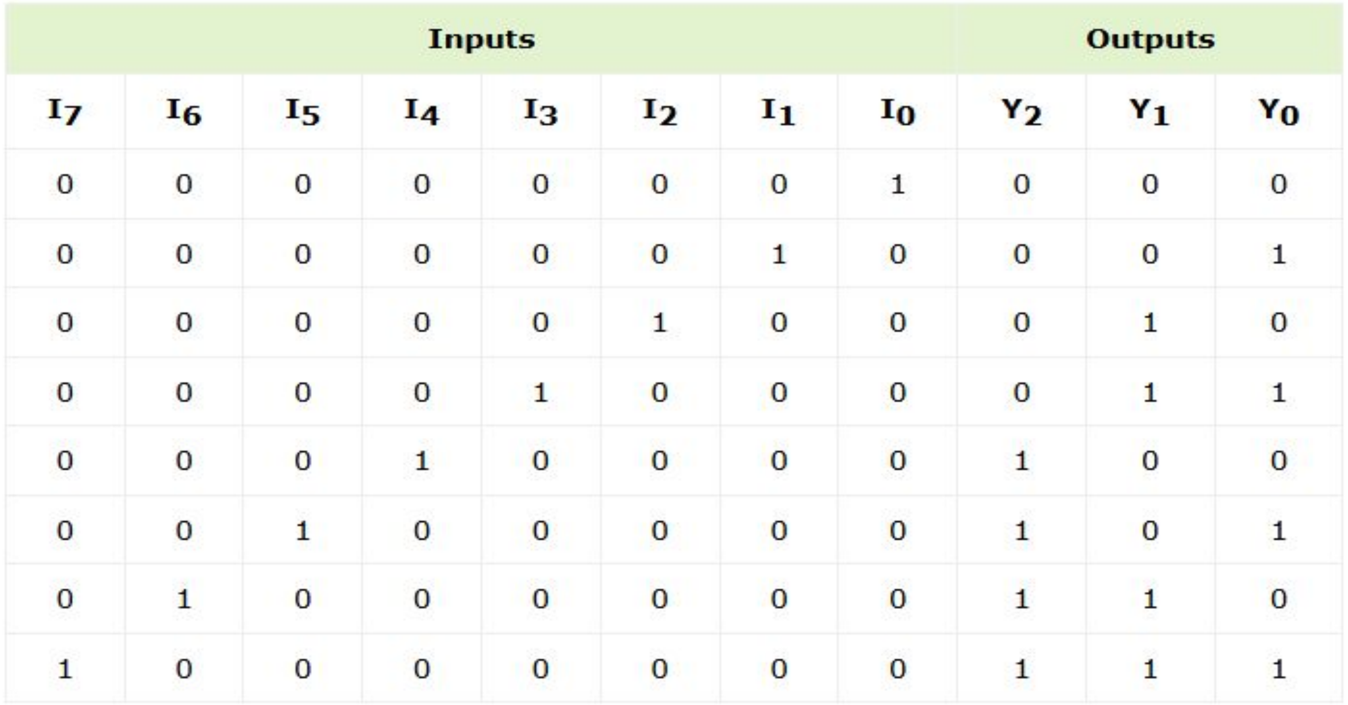
The **block diagram of an octal to binary encoder** is shown in the following figure –



Encoder

Octal to Binary Encoder

The following truth table describes the working of an octal to binary encoder −



Inputs								Outputs		
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Encoder

Octal to Binary Encoder

From this truth table, we can write the Boolean expression for the outputs of the octal to binary encoder as follows.

$$Y0 = I1 + I3 + I5 + I7$$

$$Y1 = I2 + I3 + I6 + I7$$

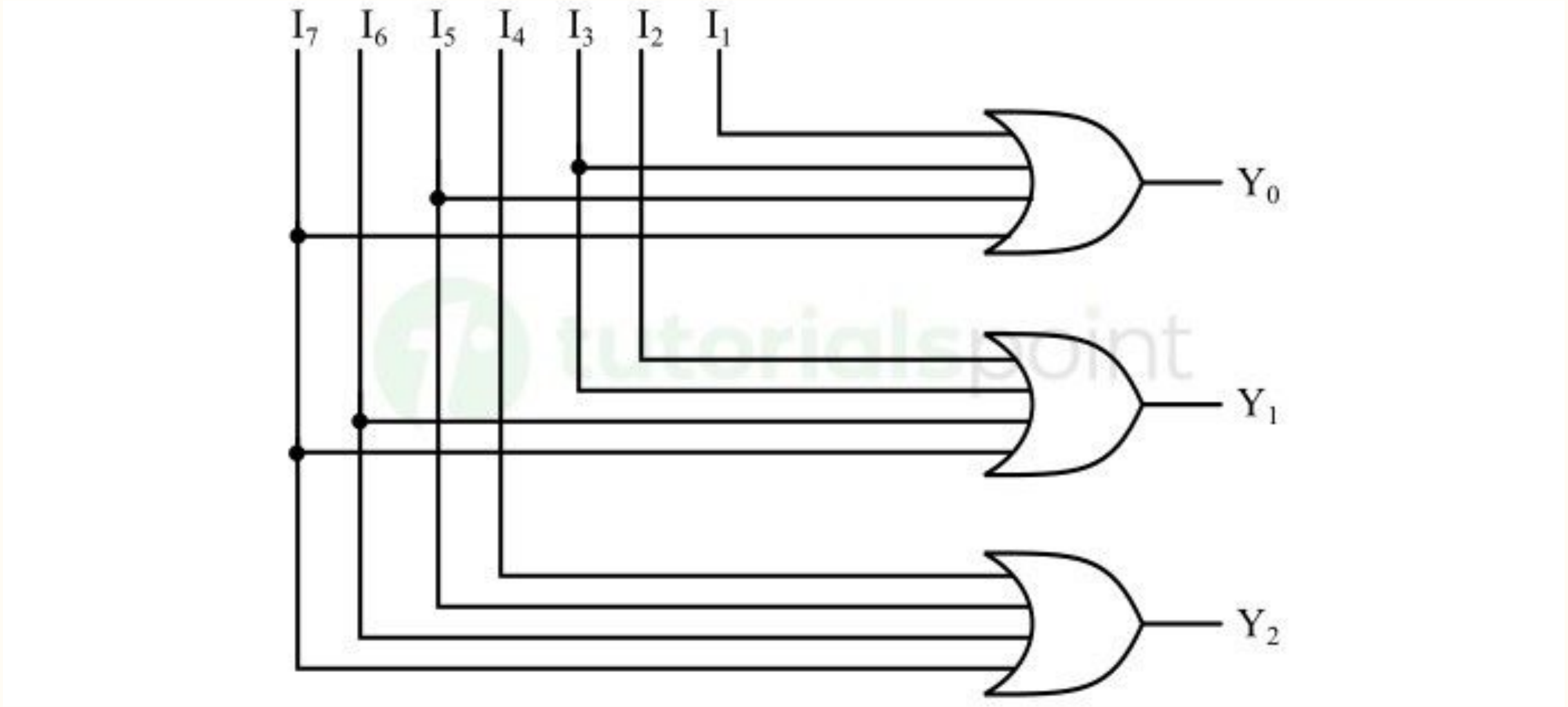
$$Y2 = I4 + I5 + I6 + I7$$

From these expressions, it is clear that the implementation of an octal to binary encoder requires 3 OR gates.

Encoder

Octal to Binary Encoder

The logic circuit diagram of the octal to binary encoder is shown in the following figure –

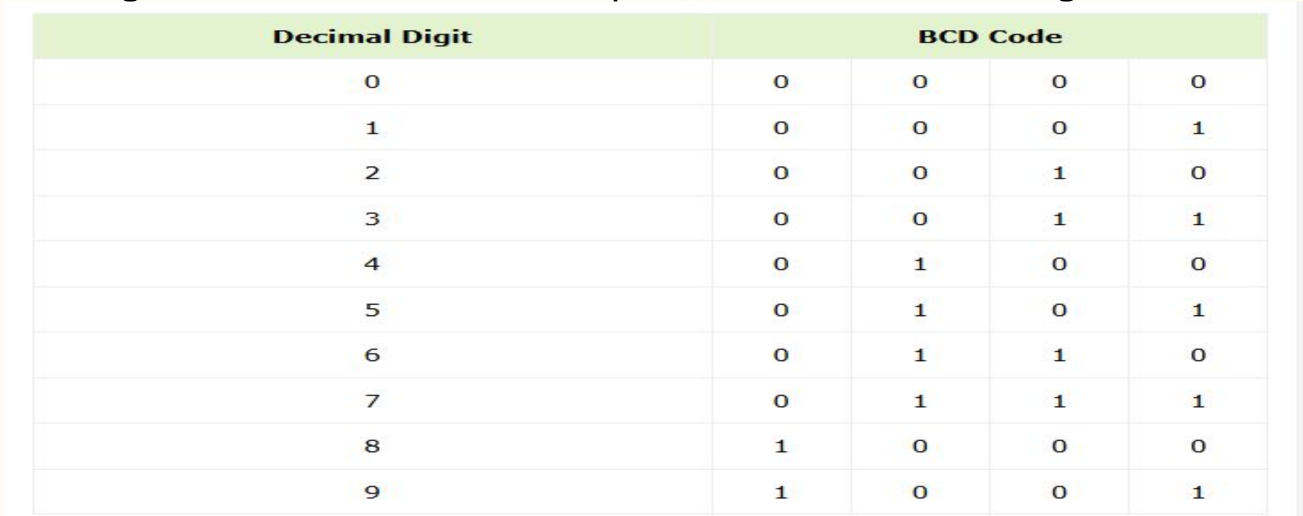


Encoder

Decimal to BCD Encoder

A type of encoder that can convert a decimal number or information represented using decimal number into its equivalent binary-coded decimal (BCD) format is known as a decimal to BCD encoder.

In the BCD encoding scheme, each decimal digit can be converted into a 4-bit binary representation. The following table shows the BCD equivalents of decimal digital from 0 to 9.



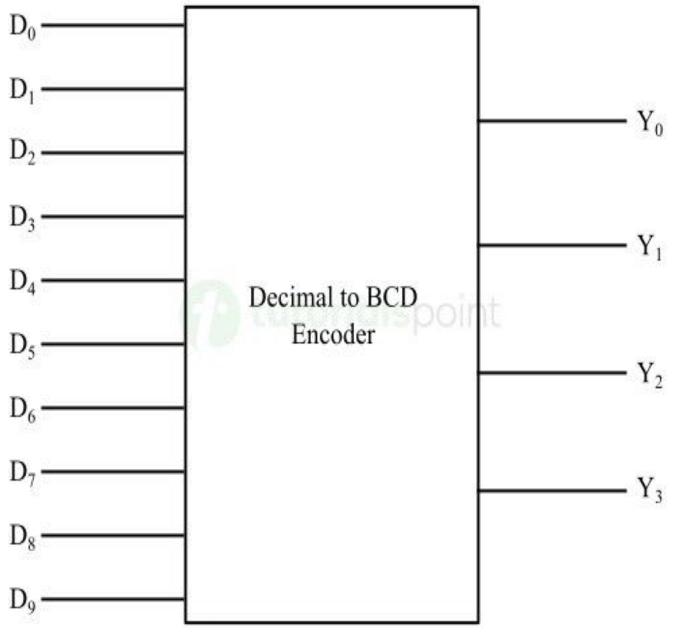
Decimal Digit	BCD Code			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Encoder

Decimal to BCD Encoder

The decimal to BCD encoder accepts 10 input lines and produces a 4-bit BCD output depending on the combination of input lines. Therefore, sometimes it is also called a 10 to 4 encoder.

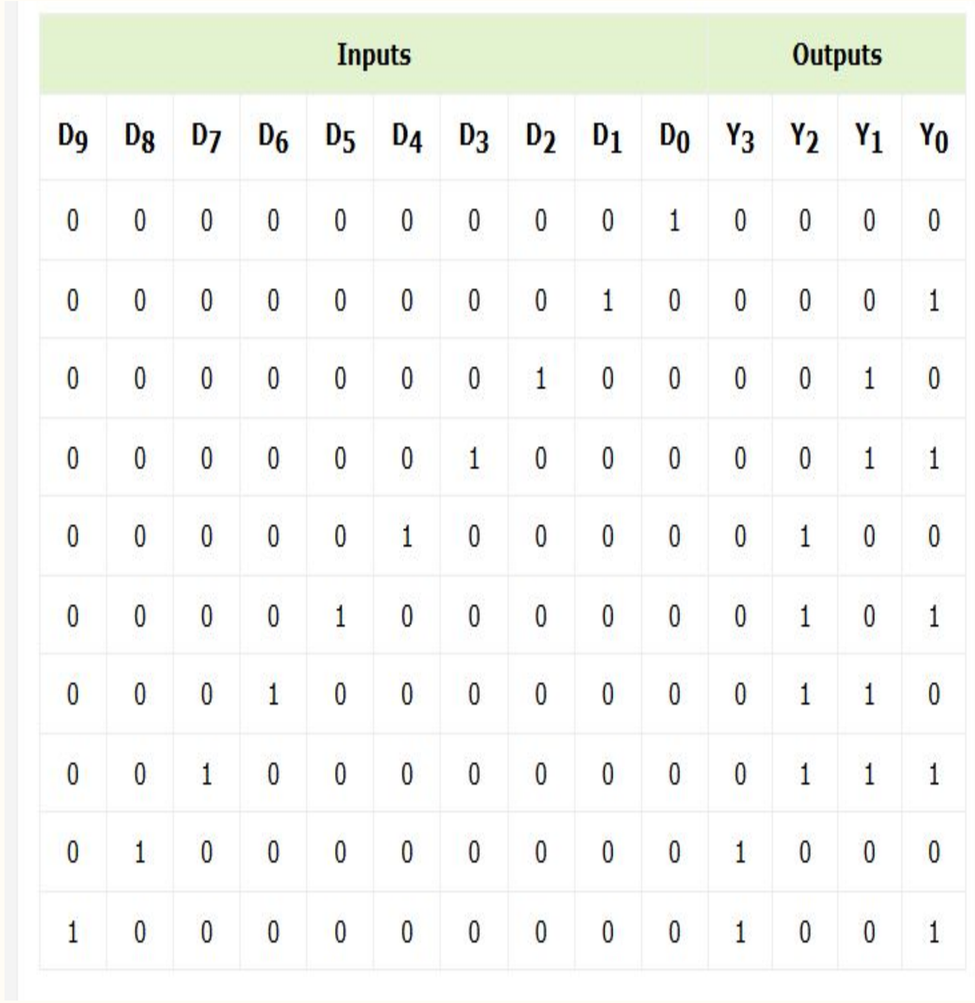
The following illustration depicts the block diagram of a decimal to BCD encoder.



Encoder

Decimal to BCD Encoder

The truth table describing the working of the decimal to BCD encoder is given blow



Inputs										Outputs			
D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

Encoder

Decimal to BCD Encoder

From this truth table, we can write the Boolean expression of the decimal to BCD encoder as follows.

$$Y_0 = D_1 + D_3 + D_5 + D_7 + D_9$$

$$Y_1 = D_2 + D_3 + D_6 + D_7$$

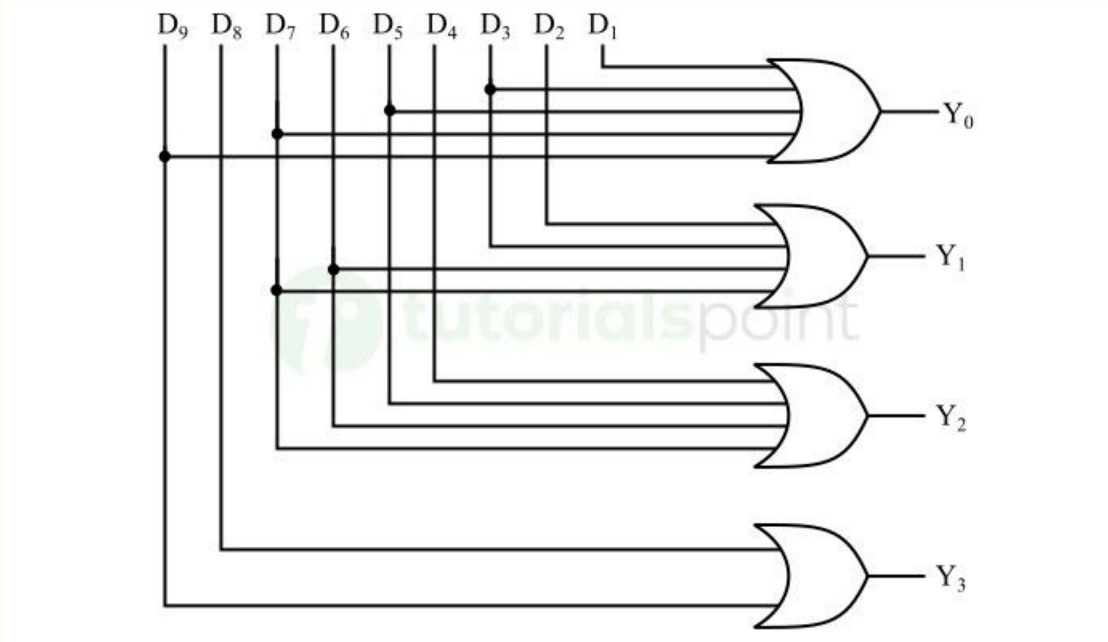
$$Y_2 = D_4 + D_5 + D_6 + D_7$$

$$Y_3 = D_8 + D_9$$

Encoder

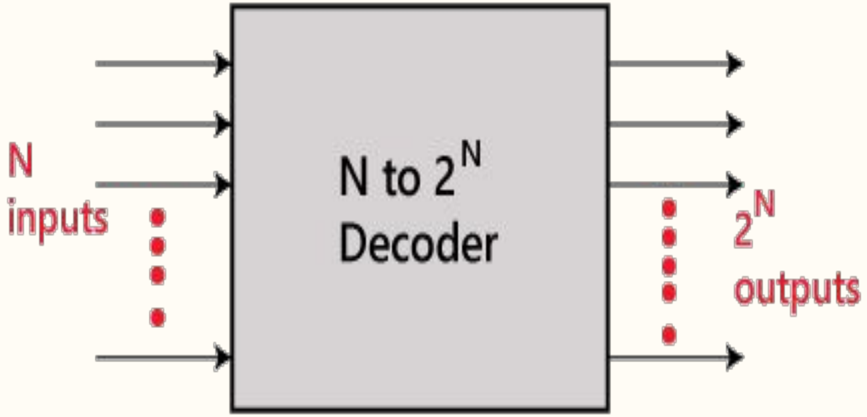
Decimal to BCD Encoder

The **logic circuit of the decimal to BCD encoder** can be implemented using four OR gates which is shown in the following figure –



Decoder

The combinational circuit that change the binary information into 2N output lines is known as Decoders. The binary information is passed in the form of N input lines. The output lines define the 2N-bit code for the binary information. In simple words, the Decoder performs the reverse operation of the Encoder. At a time, only one input line is activated for simplicity. The produced 2N-bit output code is equivalent to the binary information.



Decoder

Some of the commonly used types of Decoder in digital electronics −

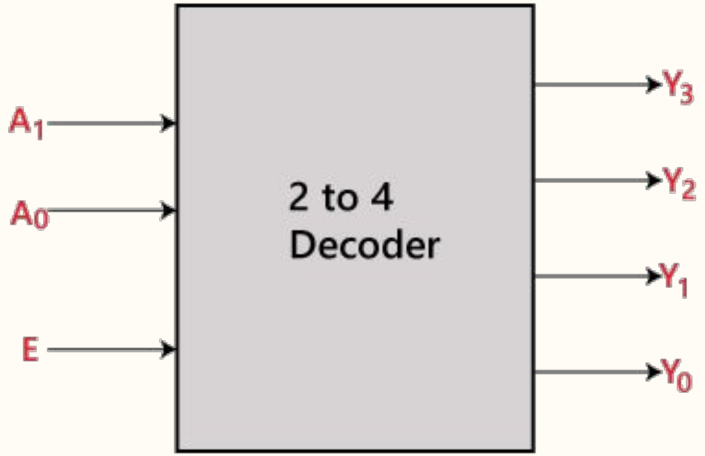
- 2 to 4 Decoder
- 3 to 8 Decoder (Octal Decoder)
- BCD to Decimal Decoder
- 4 to 16 Decoder

Let us now discuss these four types of most commonly used Decoder in detail.

Decoder

2 to 4 Decoder (Binary to decimal decoder)

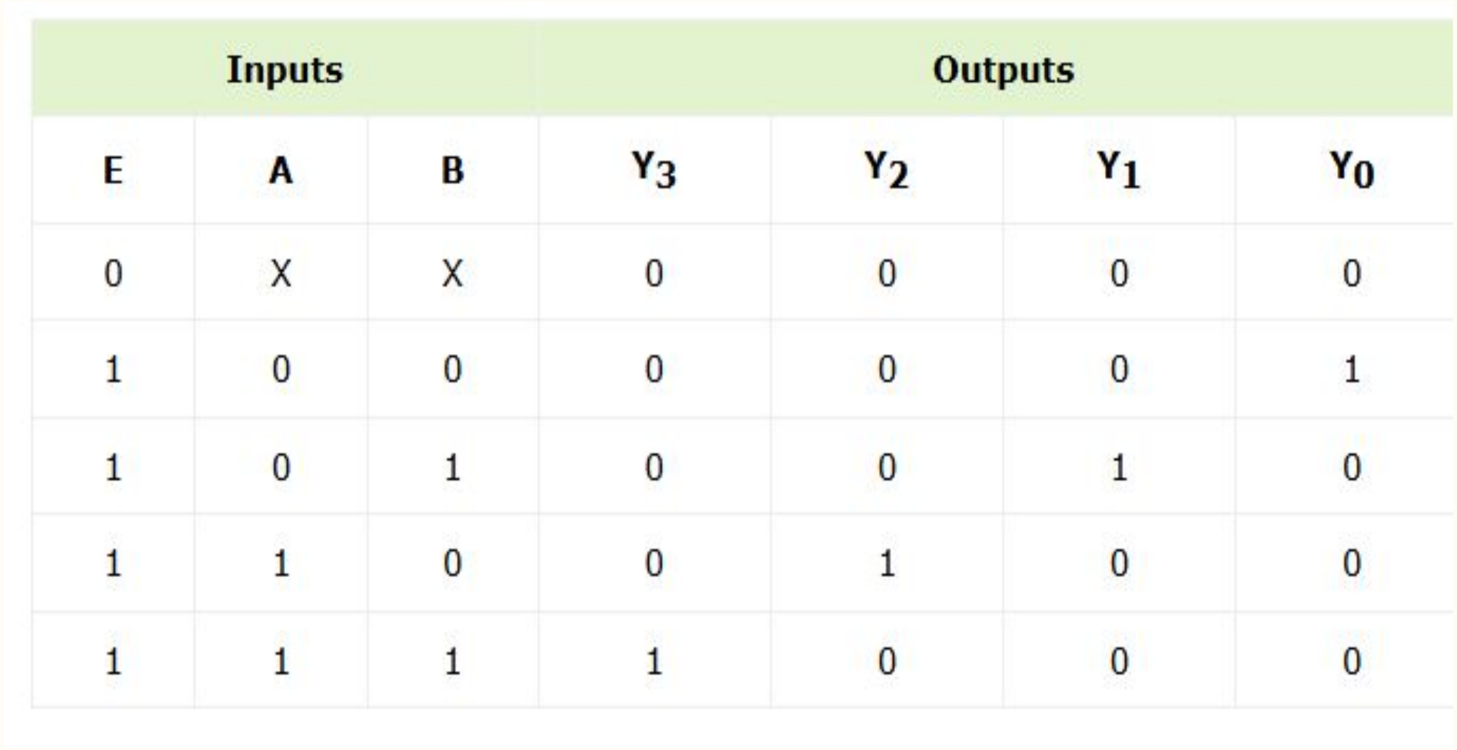
In the 2 to 4 line decoder, there is a total of three inputs, i.e., A, and B and E and four outputs, i.e., Y0, Y1, Y2, and Y3. For each combination of inputs, when the enable 'E' is set to 1, one of these four outputs will be 1. The block diagram and the truth table of the 2 to 4 line decoder are given below.



Decoder

2 to 4 Decoder (Binary to decimal decoder)

Truth Table:



Inputs			Outputs			
E	A	B	Y_3	Y_2	Y_1	Y_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Decoder

2 to 4 Decoder (Binary to decimal decoder)

The logical expression of the term Y0, Y1, Y2, and Y3 is as follows:

Y_0 = E. B′ . A′
Y_1 = E. B′ . A
Y_2 = E. B . A′
Y_3 = E. B. A

Logical circuit of the above expressions is given below:

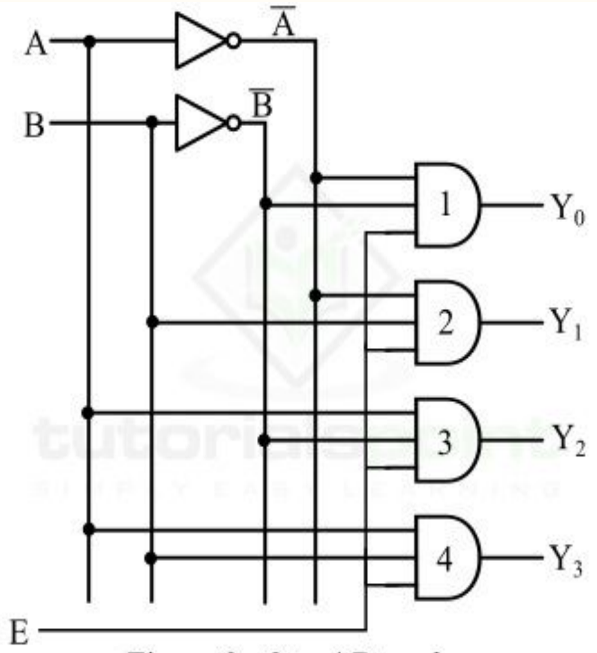


Figure 3 - 2 to 4 Decoder

Decoder

3 to 8 Decoder(octal decoder)

The 3 to 8 line decoder is also known as Binary to Octal Decoder. In a 3 to 8 line decoder, there is a total of eight outputs, i.e., Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 and three outputs, i.e., A, B, and C. This circuit has an enable input 'E'. Just like 2 to 4 line decoder, when enable 'E' is set to 1, one of these four outputs will be 1. The block diagram and the truth table of the 3 to 8 line decoder are given below.

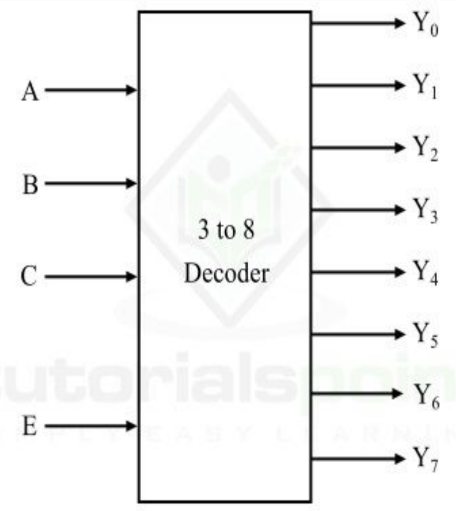
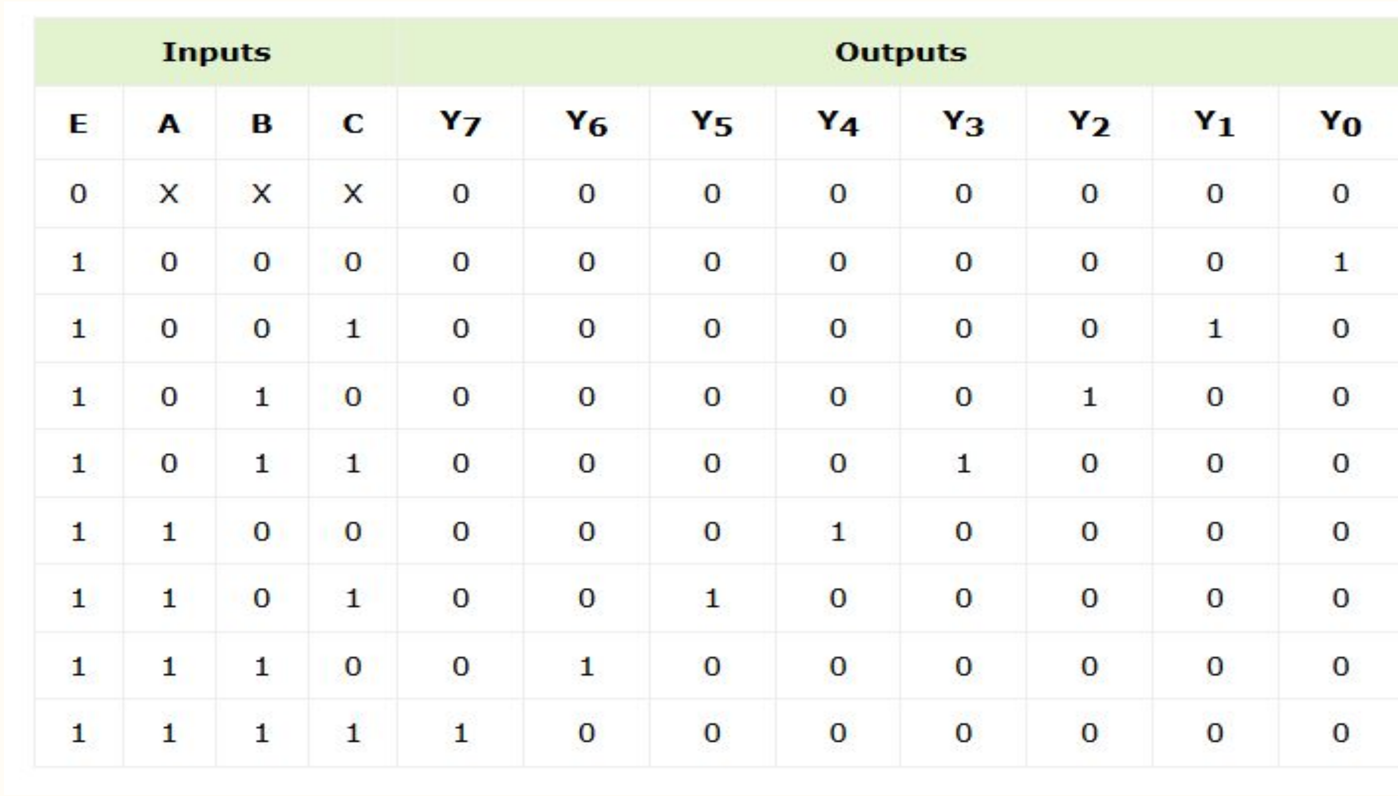


Figure 4 - 3 to 8 Decoder

Decoder

3 to 8 Decoder(octal decoder)

Truth table:



Inputs				Outputs							
E	A	B	C	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Decoder

3 to 8 Decoder(octal decoder)

The logical expression of the term Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 is as follows:

$Y_0 = E.\ A'.\ B'\ .\ C'$

$Y_1 = E.\ A'.B'.C$

$Y_2 = E.\ A'.B.C'$

$Y_3 = E.\ A'.B.C$

$Y_4 = E.\ A.\ B'.C'$

$Y_5 = E.\ A.B'.C$

$Y_6 = E.\ A.B.C'$

$Y_7 = E.\ A.B.C$

Decoder

3 to 8 Decoder(octal decoder)

Logical Diagram:

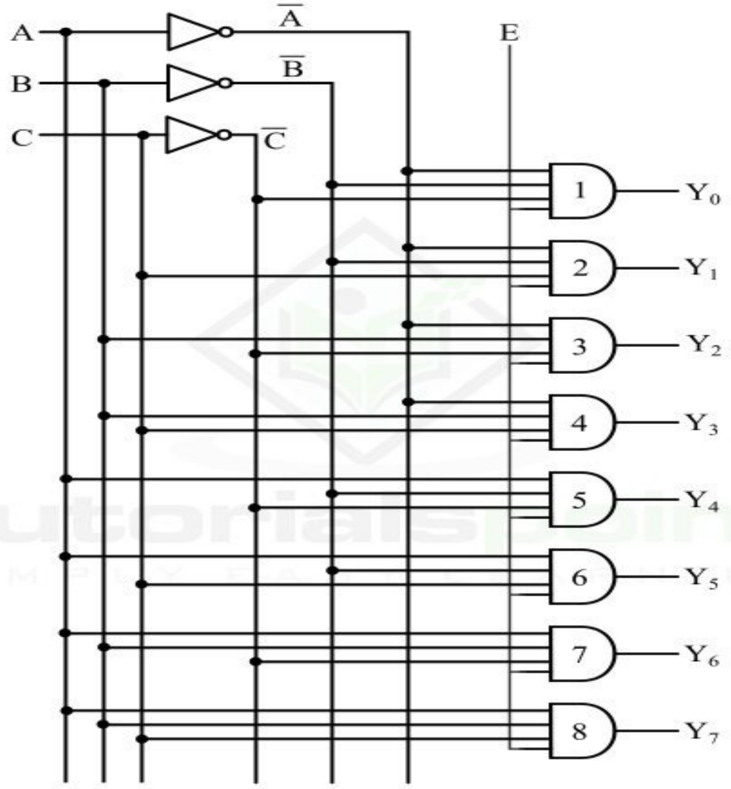


Figure 5 - 3 to 8 Decoder

Decoder

BCD to decimal decoder

A digital circuit that can convert a binary-coded decimal (BCD) number into an equivalent decimal number is referred to as a **BCD-to-decimal converter**.

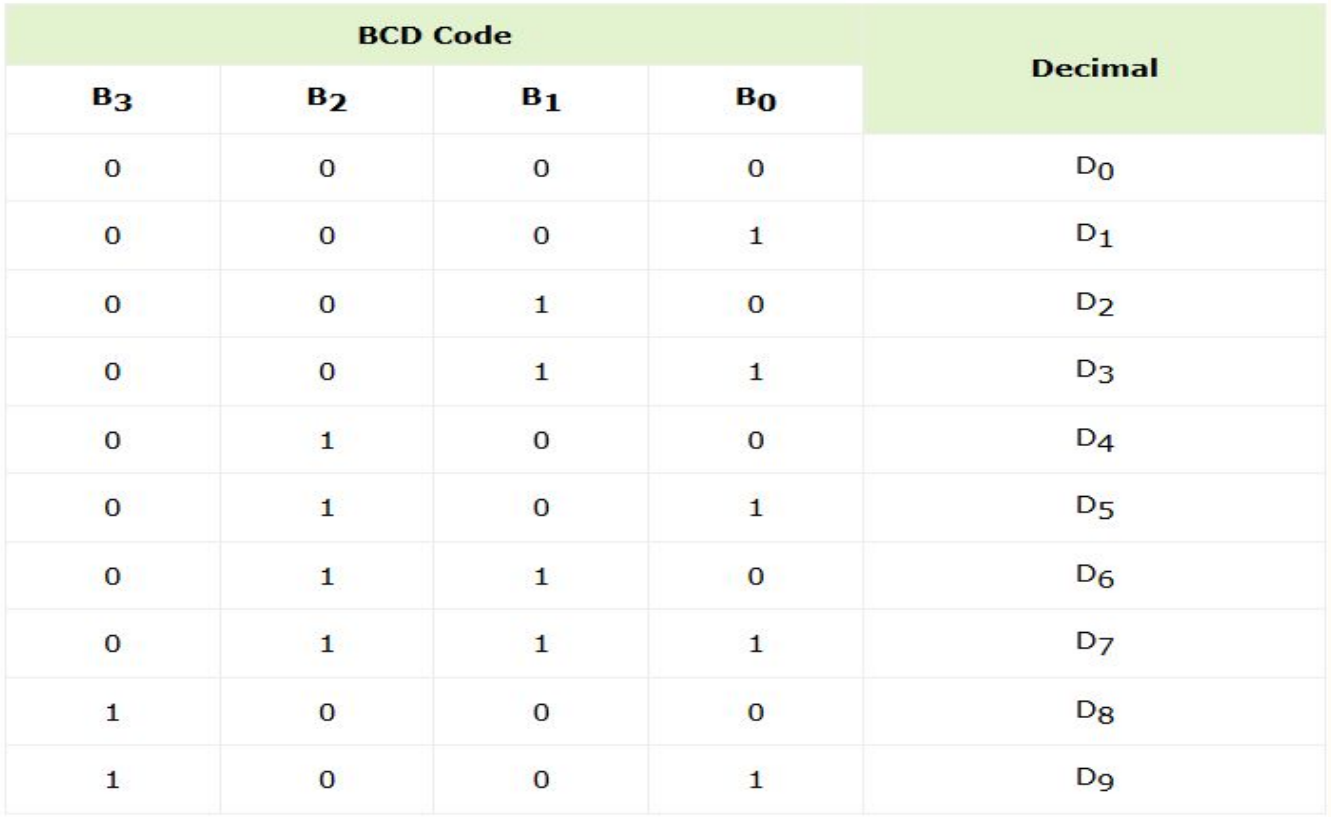
The input to a BCD to decimal converter is an 8421 BCD code and the output generated by the converter is a decimal number.

The following is the truth table of the BCD to decimal converter describing its operation.

Decoder

BCD to decimal decoder

Truth table:



BCD Code				Decimal
B_3	B_2	B_1	B_0	
0	0	0	0	D_0
0	0	0	1	D_1
0	0	1	0	D_2
0	0	1	1	D_3
0	1	0	0	D_4
0	1	0	1	D_5
0	1	1	0	D_6
0	1	1	1	D_7
1	0	0	0	D_8
1	0	0	1	D_9

Decoder

BCD to decimal decoder

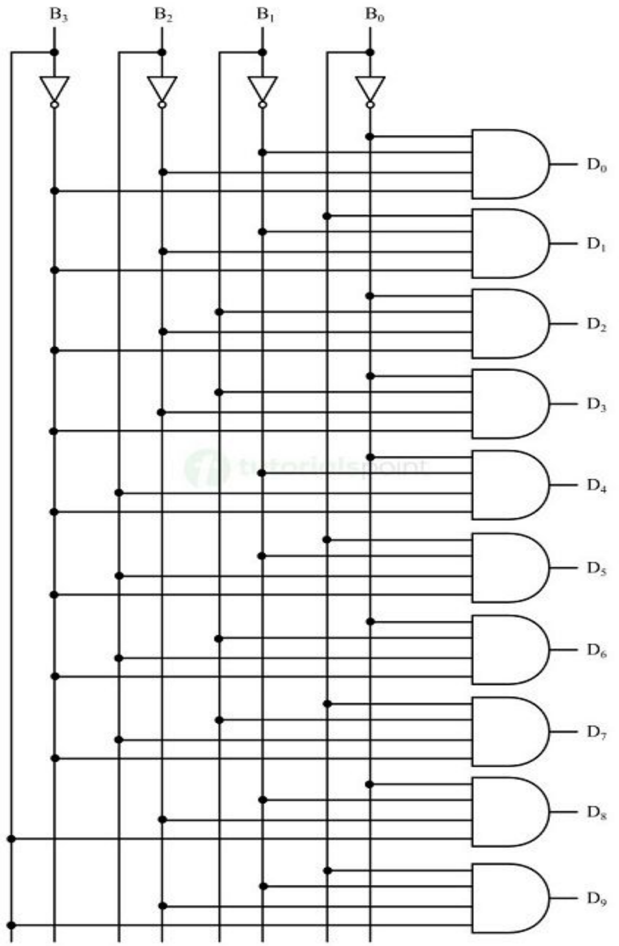
We can derive the Boolean expressions for each of the decimal outputs in terms of 8421 BCD code. These Boolean expressions are given below −

$$D_0 = B_3'. B_2'. B_1' . B_0'$$
$$D_1 = B_3'. B_2' . B_1'. B_0$$
$$D_2 = B_3'. B_2' . B_1. B_0'$$
$$D_3 = B_3'. B_2' . B_1 . B_0$$
$$D_4 = B_3'. B_2 . B_1'. B_0'$$
$$D_5 = B_3'. B_2 . B_1'. B_0$$
$$D_6 = B_3'. B_2 . B_1 . B_0'$$
$$D_7 = B_3'. B_2 . B_1 . B_0$$
$$D_8 = B_3 . B_2'. B_1'. B_0'$$
$$D_9 = B_3 . B_2'. B_1'. B_0$$

Decoder

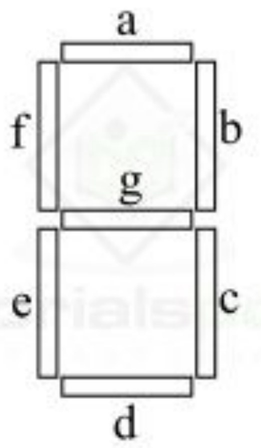
BCD to decimal decoder

The logic circuit implementation of the BCD to decimal converter is shown in the following figure.



Decoder

7 Segment LED Display Decoder

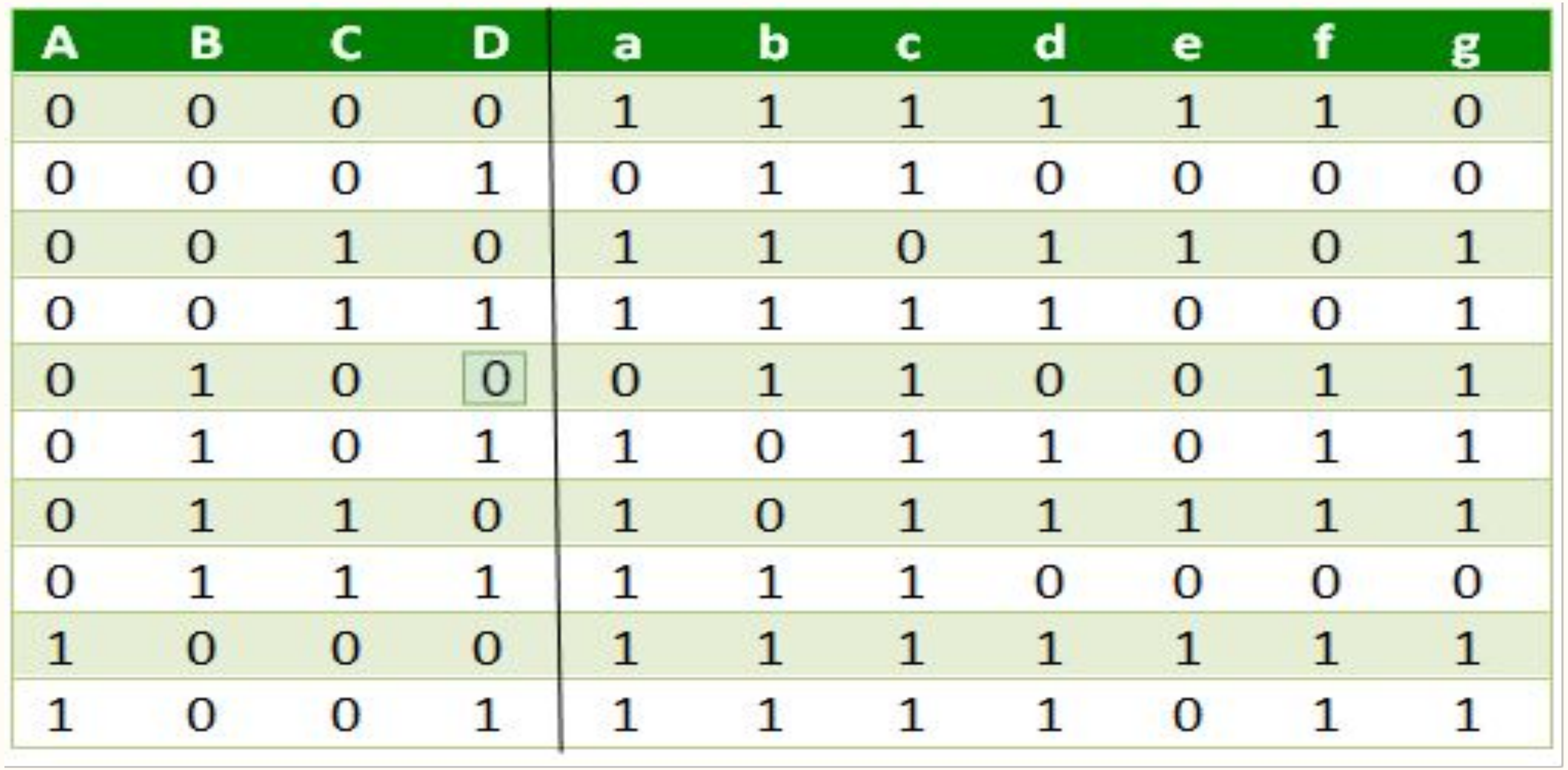


A seven segment decoder is the combinational circuit that take BCD input and give the output in seven segment display. It is the combinational of 7 segment LED. The following is the truth table for the seven segment decoder.

Decoder

7 Segment LED display decoder

Truth table:



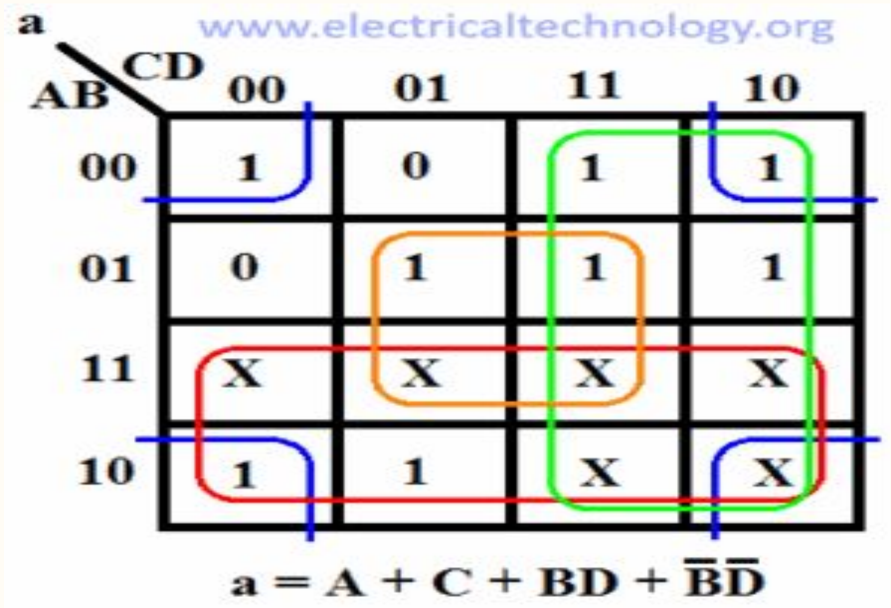
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

Decoder

7 Segment LED display decoder k map

For other combinations of input, the output is “don’t care X” as there are no more digits to display. We will derive the expression for each output using k-map

For output a:



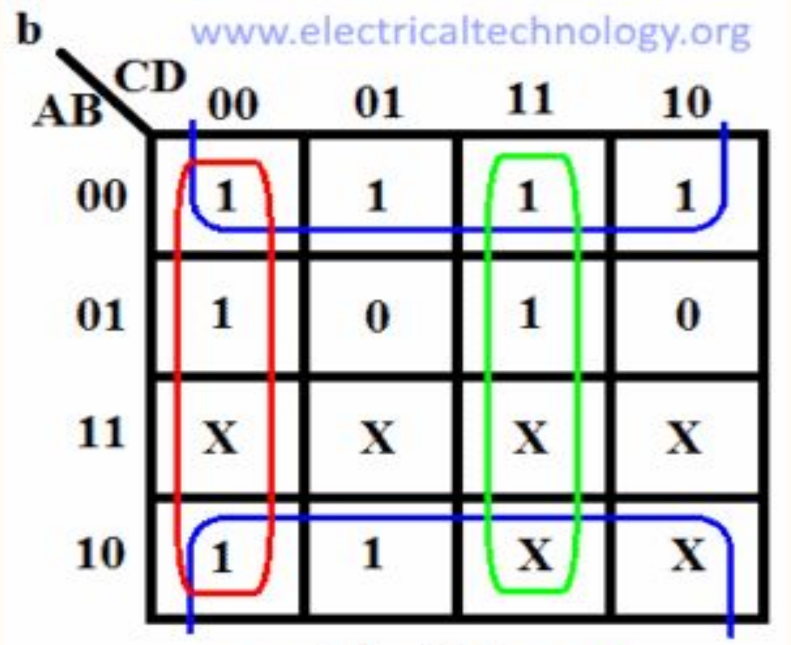
a: AB \ CD	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

$$a = A + C + BD + \overline{B}\,\overline{D}$$

Decoder

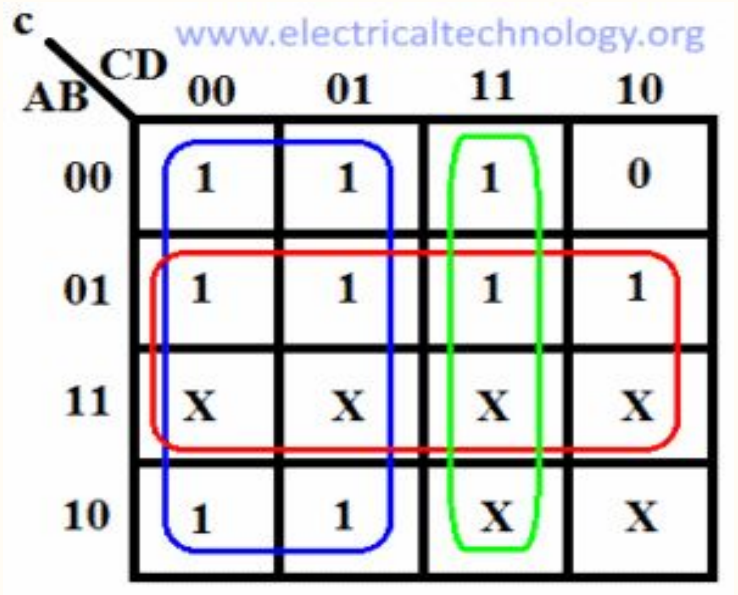
7 Segment LED display decoder k map

For output b and c:



AB \ CD	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	X	X	X	X
10	1	1	X	X

$b = \overline{B} + \overline{C}\,\overline{D} + CD$



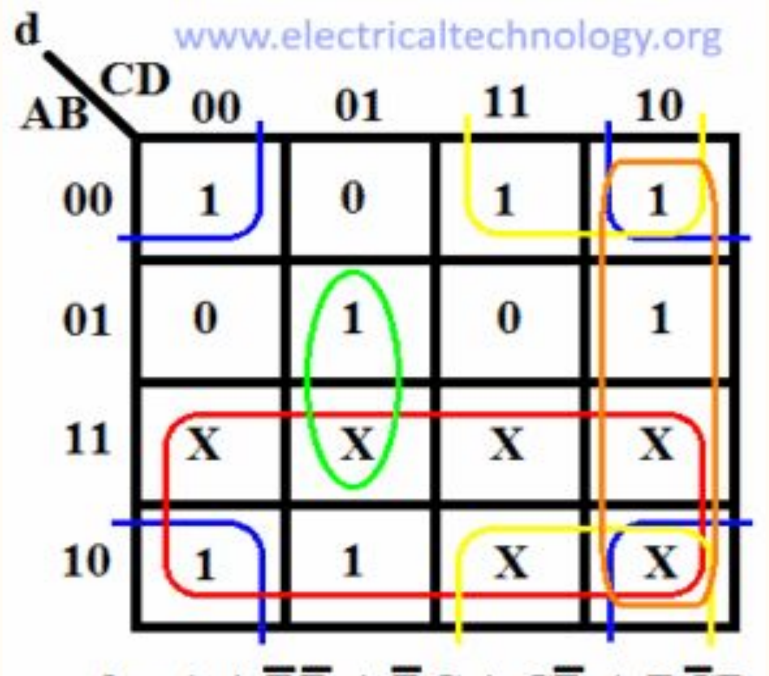
AB \ CD	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

$c = B + \overline{C} + D$

Decoder

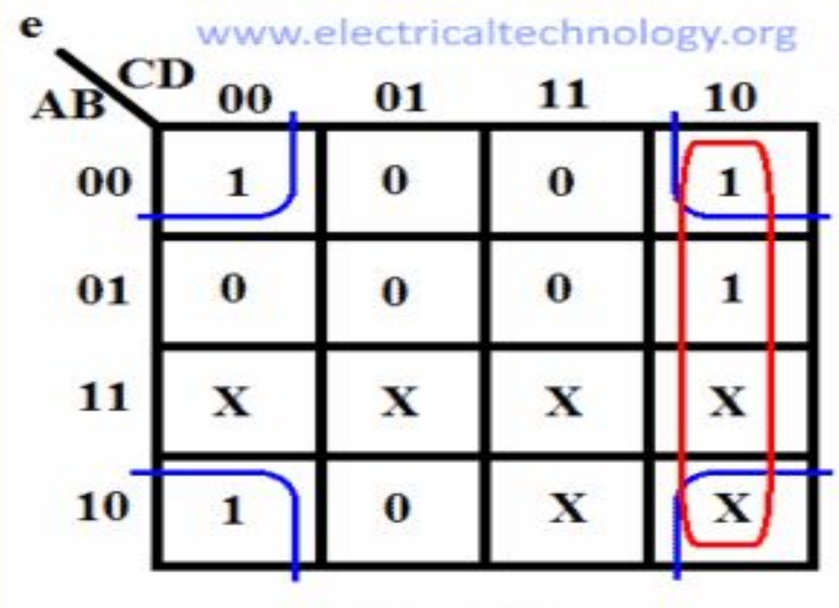
7 Segment LED display decoder k map

For output d and e:



d: AB \ CD	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	X	X	X	X
10	1	1	X	X

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



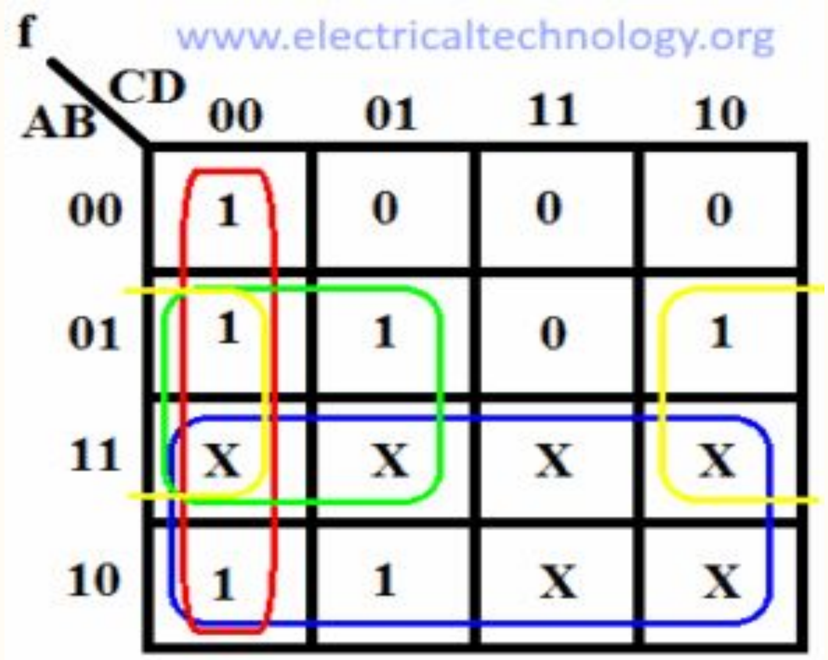
e: AB \ CD	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	X	X	X	X
10	1	0	X	X

$e = \overline{B}\overline{D} + C\overline{D}$

Decoder

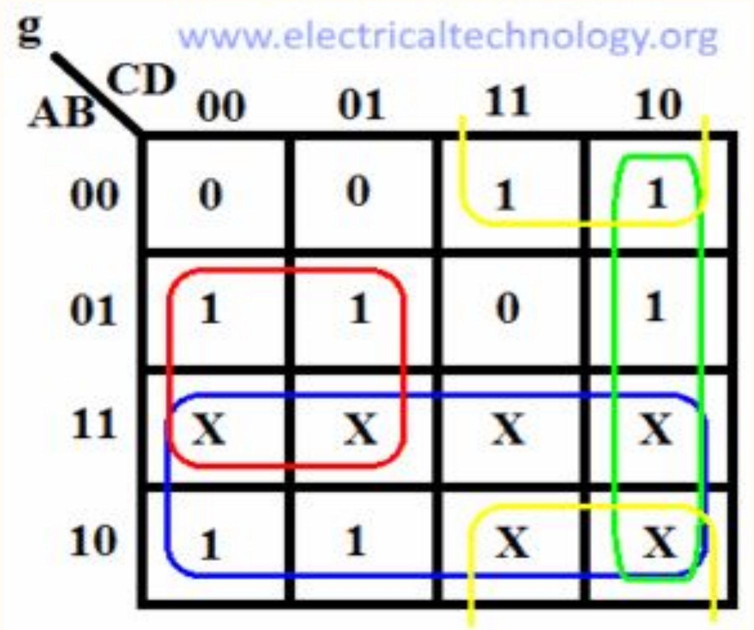
7 Segment LED display decoder k map

For output f and g:



AB \ CD	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

$$f = A + B\overline{C} + B\overline{D} + \overline{C}\,\overline{D}$$



AB \ CD	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

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

Multiplexer

A multiplexer (MUX)is a digital combinational logic circuit that accepts multiple data inputs and allows only one of them at a time to transmit over an output channel. A multiplexer consists of 2^n input lines, n select lines, and one output line.

In a multiplexer, the routing of the desired data input to the output channel is done by the logic level applied to the select lines. The functional block diagram of a multiplexer is shown

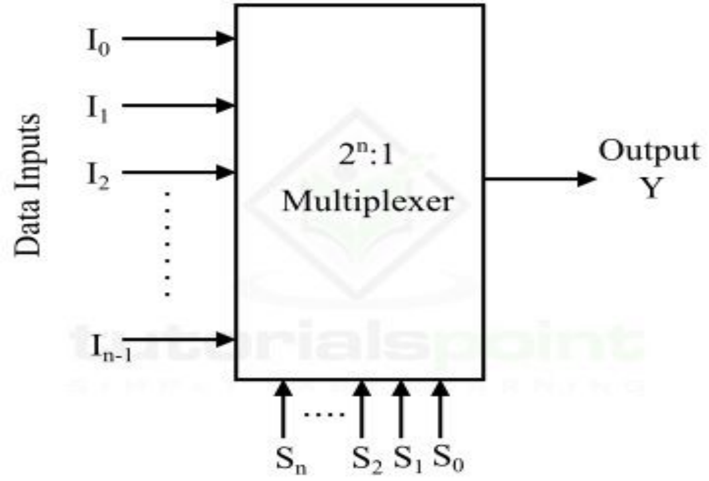


Figure 1 - Digital Multiplexer

Multiplexer

A multiplexer has categories into three types . They are :

2:1 multiplexer

4:1 multiplexer

8:1 multiplexer

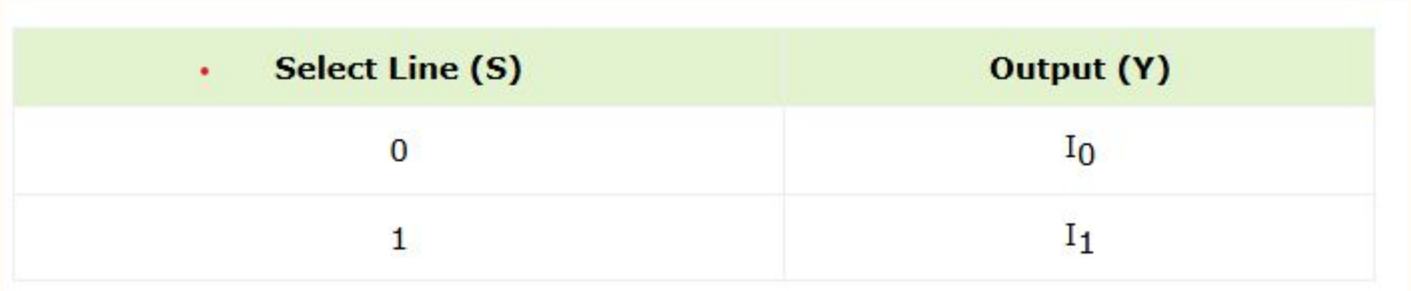
2:1 multiplexer

A 2:1 multiplexer (MUX) is one which has 2 input lines (I_0 and I_1), 1 select line (S), and 1 output line (Y). The logic level applied to the select line S determines which data input will pass through the output line.

For determining the Boolean expression for output (Y) of the 2:1 multiplexer and its logic circuit implementation, we first need its function table (truth table) that gives information about operation of the circuit. The function table of the 2:1 multiplexer with data input I**0** and I**1** is shown below.

Multiplexer

2:1 multiplexer



Select Line (S)	Output (Y)
0	I_0
1	I_1

Using this truth table, we can write the logic expression for the output of 2:1 MUX as,

$$Y=S'I_0 + SI_1$$

To implement this logic expression, we require two AND gates, one NOT gate, and one OR gate. The logic circuit of the 2:1 MUX is shown.

Multiplexer

2:1 multiplexer

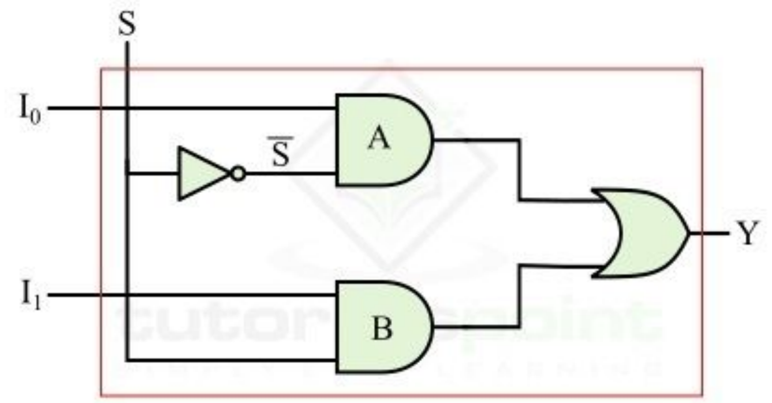
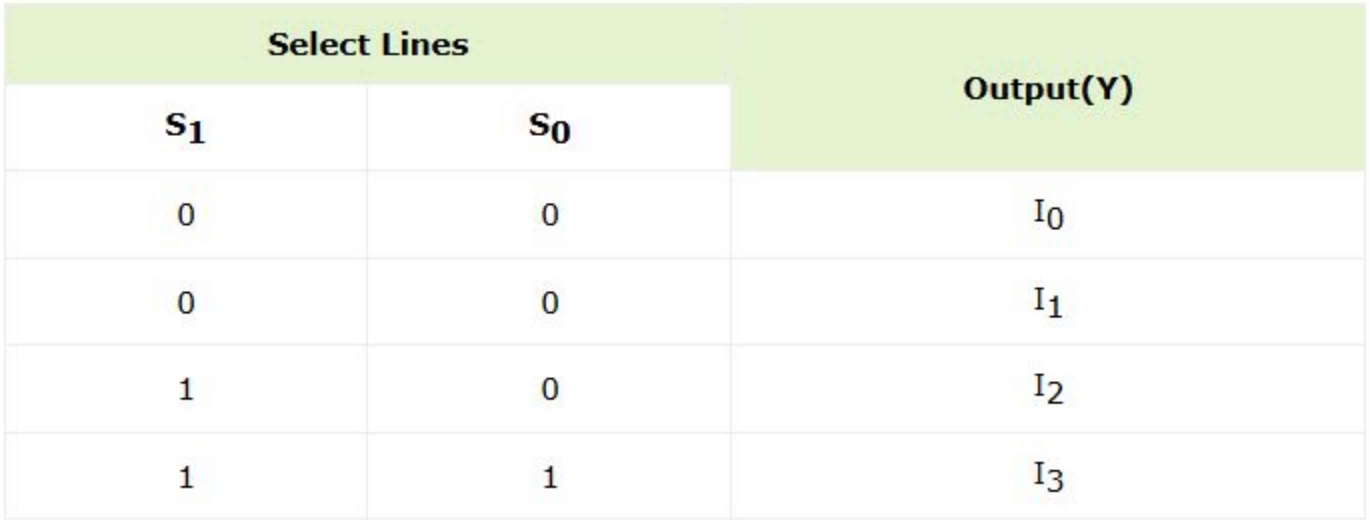


Figure 2 - 2:1 Multiplexer

Multiplexer

4:1 multiplexer

A 4:1 multiplexer (MUX) is the MUX which has 4 input lines (I_0, I_1, I_2, and I_3), 2 select line (S_0 and S_1), and 1 output line (Y). The logic levels applied to the select lines S_0 and S_1 determine which data input will transmit to the output line.



Select Lines		Output(Y)
S_1	S_0	
0	0	I_0
0	0	I_1
1	0	I_2
1	1	I_3

Multiplexer

4:1 multiplexer

Using this truth table, we can write the logic expression for the output of 4:1 MUX as,

$$E= S'_1S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$$

To implement this logic expression, we require four AND gates, two NOT gates, and one OR gate. Thus, the logic circuit of the 4:1 MUX is shown

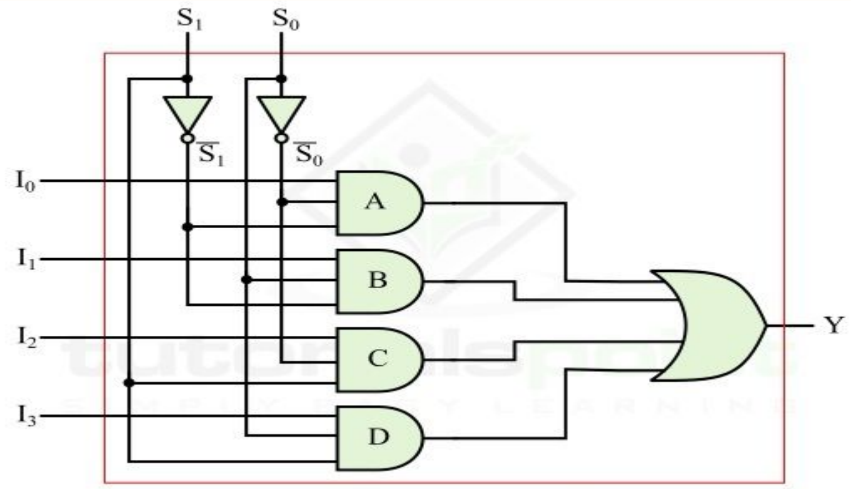
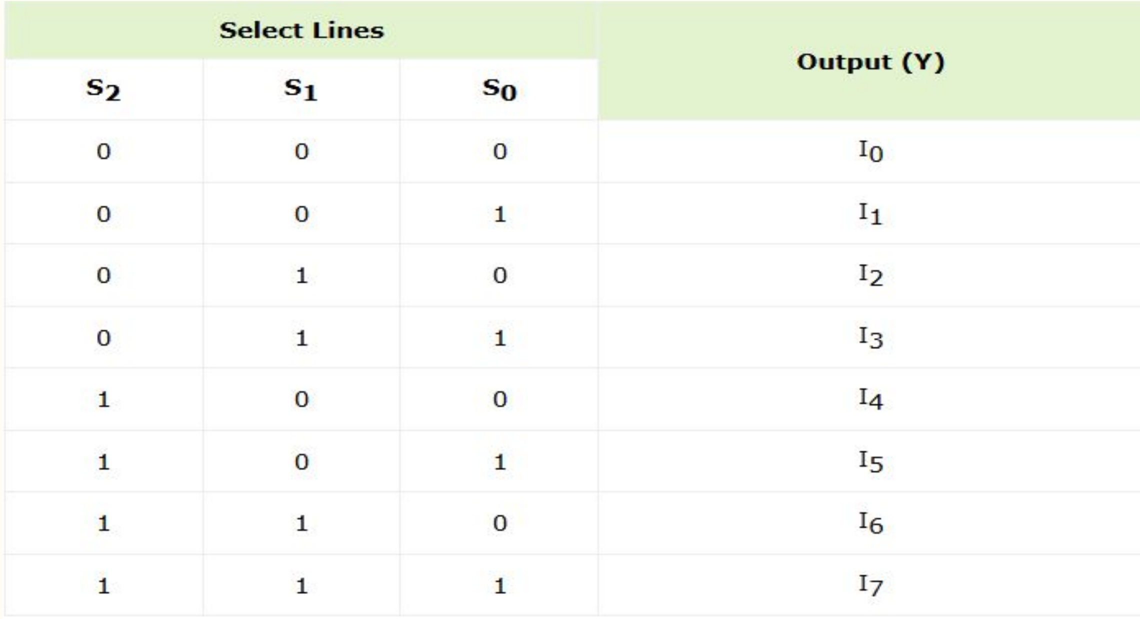


Figure 3 - 4:1 Multiplexer

Multiplexer

8:1 multiplexer

An 8:1 multiplexer (MUX) is a combinational logic circuit which has 8 input lines (I_0, I_1, I_2, I_3, I_4, I_5, I_6, and I_7), 3 select line (S_0, S_1, and S_2), and 1 output line (Y). The logic levels applied to the select lines S_0, S_1, and S_2 determine which data input will transmit to the output line.



Select Lines			Output (Y)
S_2	S_1	S_0	
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

Multiplexer

8:1 multiplexer

By using this function table, we can write the logic expression for the output of 8:1 MUX as,

$Y=S_2'S_1'S_0'I_0+S_2'S_1'S_0I_1+S_2'S_1S_0'I_2+S_2'S_1S_0I_3+S_2S_1'S_0'I_4+S_2S_1'S_0I_5+S_2S_1S_0'I_6+S_2S_1S_0I_7$

To implement this logic expression, we require eight AND gates, three NOT gates, and one OR gate. Therefore, the logic circuit of the 8:1 MUX is shown

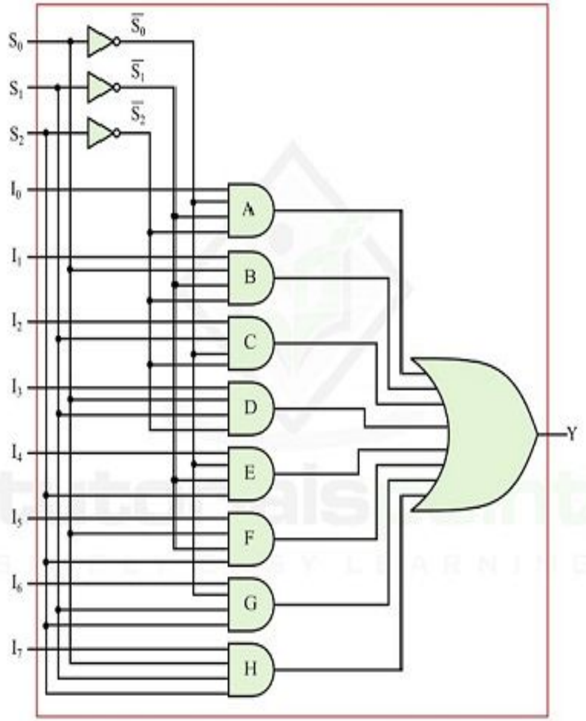


Figure 4 - 8:1 Multiplexer

Demultiplexer

A Demultiplexer is a combinational logic circuit that accepts a single input and distributes it over several output lines. Demultiplexer is also termed as DEMUX in short. As Demultiplexer is used to transmit the same data to different destinations, hence it is also known as data distributor.

There is another combinational logic circuit named multiplexer which performs opposite operation of the Demultiplexer, i.e. accepts several inputs and transmits one of them at time to the output line.

we can state that a Demultiplexer is a 1-to-2^n device. The functional block diagram of a typical 1×2^n Demultiplexer is shown

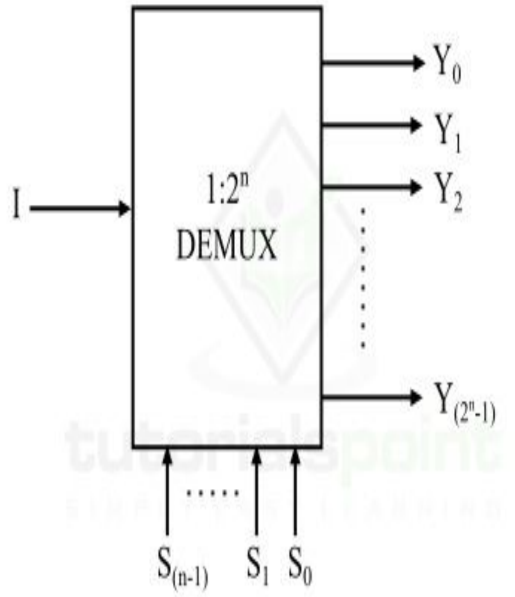


Figure 1 - Demultiplexer

Demultiplexer

Types of Demultiplexer

Based on the number of output lines (2^n), Demultiplexers can be classified into several types. Some commonly used types of De-multiplexers are −

1 to 4 demultiplexer

1 to 8 demultiplexer

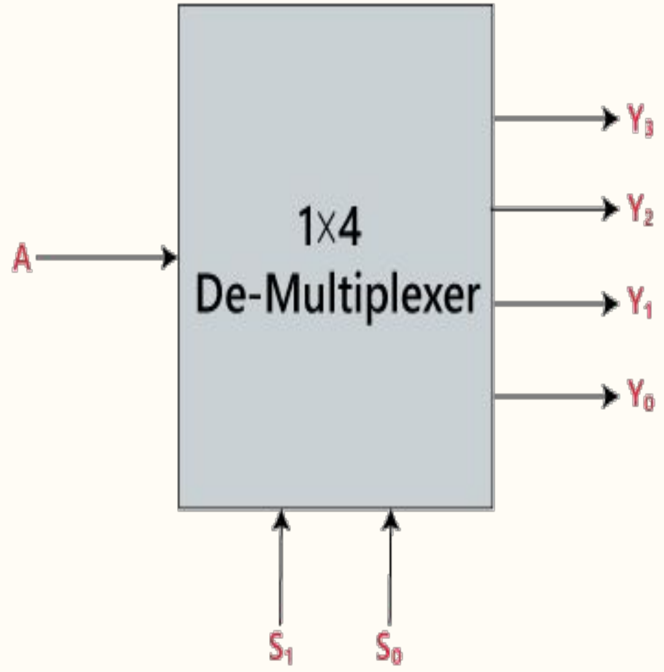
1 to 16 demultiplexer

Demultiplexer

Types of Demultiplexer

1 to 4 demultiplexer

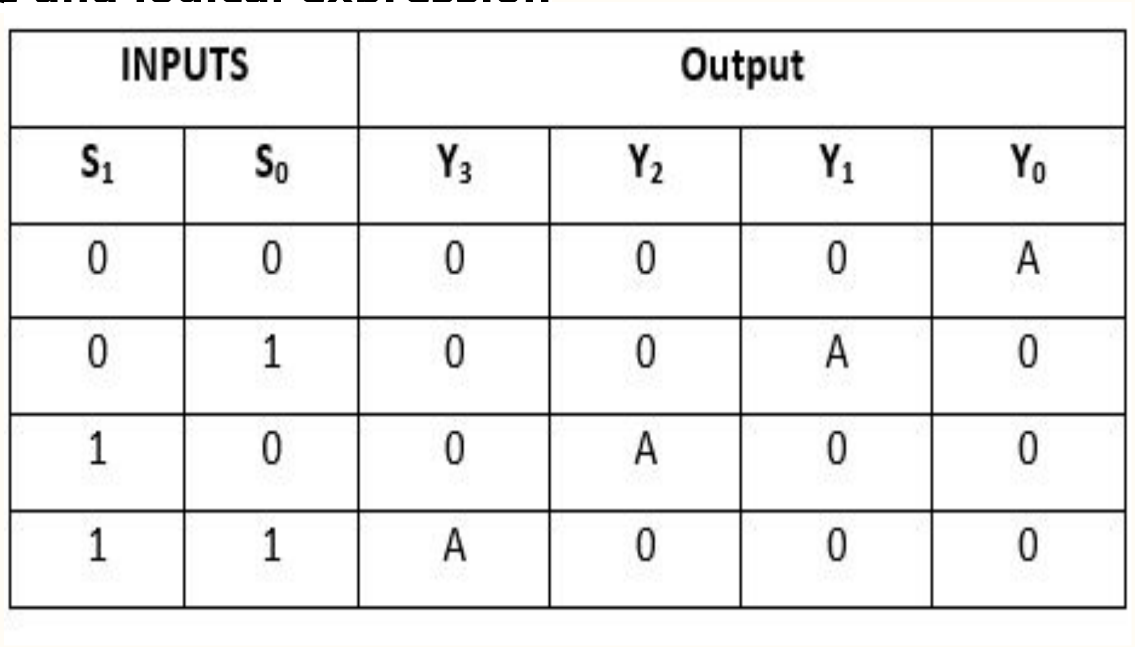
In 1 to 4 De-multiplexer, there are total of four outputs, i.e., Y0, Y1, Y2, and Y3, 2 selection lines, i.e., S0 and S1 and single input, i.e., A. On the basis of the combination of inputs which are present at the selection lines S0 and S1, the input be connected to one of the outputs. The block diagram and the truth table of the 1×4 multiplexer are given below.



Demultiplexer

Types of Demultiplexer

1 to 4 demultiplexer Truth table and logical expression



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

Logical expression

Y0=S1' S0' A

y1=S1' S0 A

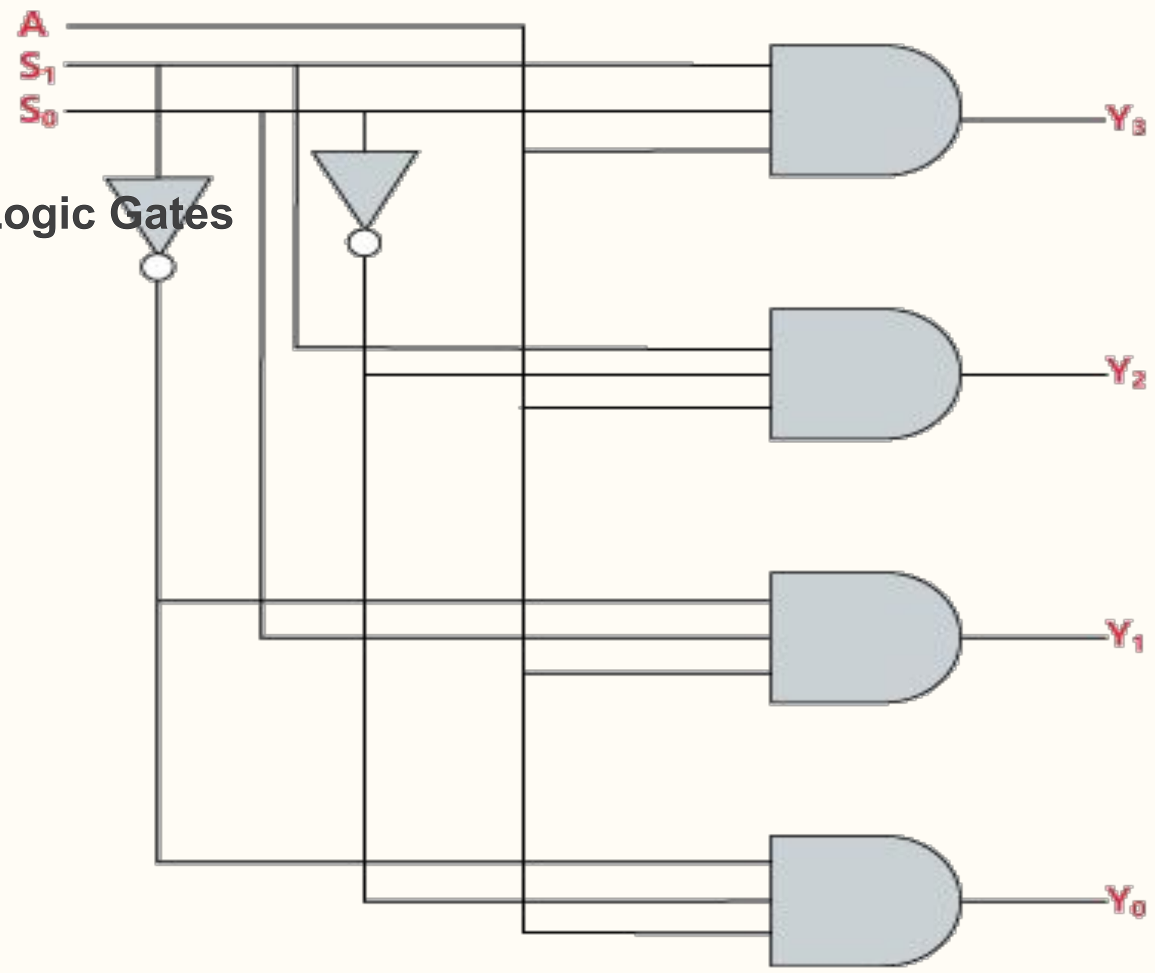
y2=S1 S0' A

y3=S1 S0 A

Demultiplexer

Types of Demultiplexer

4 Channel Demultiplexer using Logic Gates

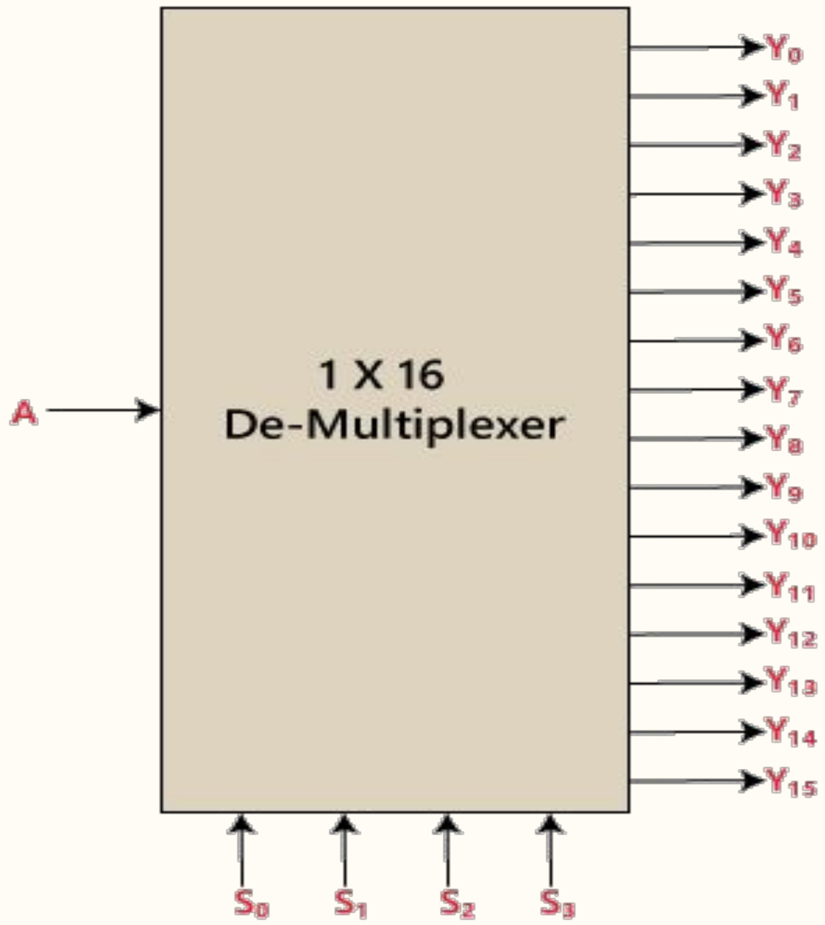


Demultiplexer

Types of Demultiplexer

1 to 16 demultiplexer

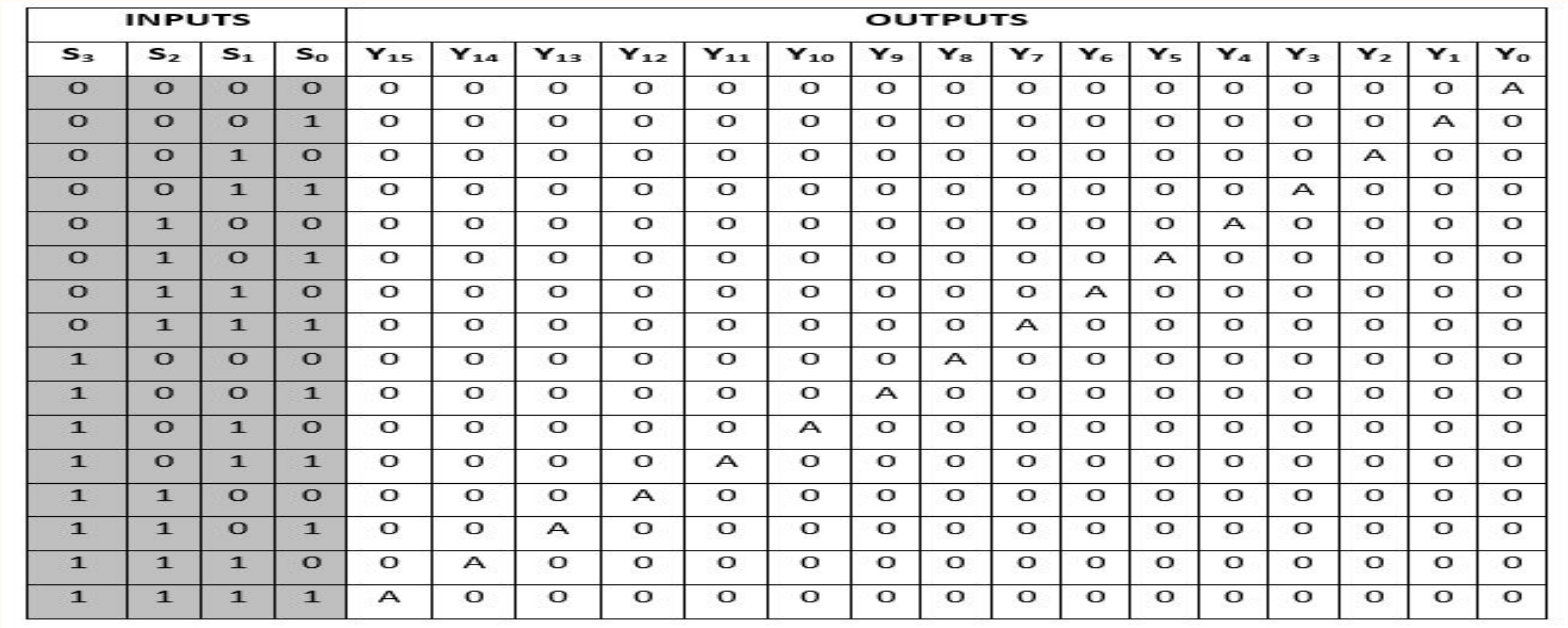
In 1×16 de-multiplexer, there are total of 16 outputs, i.e., Y0, Y1, ..., Y16, 4 selection lines, i.e., S0, S1, S2, and S3 and single input, i.e., A. On the basis of the combination of inputs which are present at the selection lines S0, S1, and S2, the input will be connected to one of these outputs. The block diagram and the truth table of the 1×16 de-multiplexer are given below



Demultiplexer

Types of Demultiplexer

1 to 16 demultiplexer Truth table



INPUTS				OUTPUTS															
S_3	S_2	S_1	S_0	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Demultiplexer

Types of Demultiplexer

1 to 16 demultiplexer logical expression

Y0=A.S0'.S1'.S2'.S3'

Y1=A.S0'.S1'.S2'.S3

Y2=A.S0'.S1'.S2.S3'

Y3=A.S0'.S1'.S2.S3

Y4=A.S0'.S1.S2'.S3'

Y5=A.S0'.S1.S2'.S3

Y6=A.S0'.S1.S2.S3'

Y7=A.S0'.S1.S2.S3

Y8=A.S0.S1'.S2'.S3'

Y9=A.S0.S1'.S2'.S3

Y10=A.S0.S1'.S2.S3'

Y11=A.S0.S1'.S2.S3

Y12=A.S0.S1.S2'.S3

Y13=A.S0.S1.S2'.S3

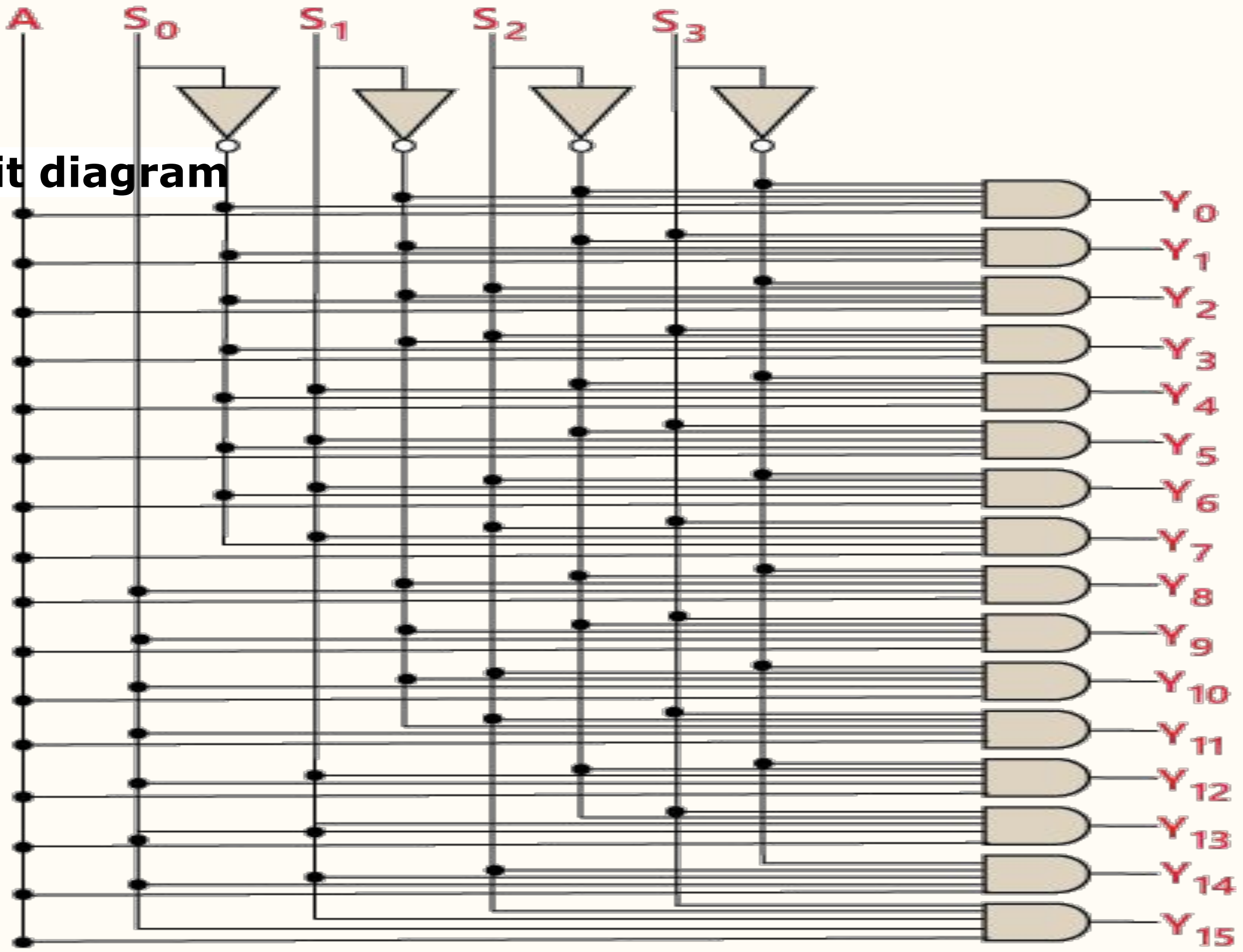
Y14=A.S0.S1.S2.S3'

Y15=A.S0.S1.S2'.S3

Demultiplexer

Types of Demultiplexer

1 to 16 demultiplexer circuit diagram



Demultiplexer and Decoder Relation

The difference between Decoder and Demultiplexer are:

Decoder	**Demultiplexer**
A decoder decodes an encrypted input signal to multiple output signals from one format to another format.	A demultiplexer routes an input signal to multiple output signals.
A decoder has input lines and a maximum of {2} ^ {n} output lines.	A demultiplexer has single input, selection lines and maximum of {2} ^ {n} outputs.
Decoder's inverse is Encoder.	Demultiplexer's inverse is Multiplexer.
Decoders are used to detect bits, encoding of data.	Demultiplexers are used in switching, data distribution.
Decoders have no select lines.	Demultiplexers contain select lines.
Decoders are heavily used in networking applications.	Demultiplexers are employed in communication systems.