

# COMBINATIONAL CIRCUITS

## Difference between combinational and sequential circuit

Last Updated: 12-09-2018

Prerequisite – [Combinational circuits using Decoder](#), [Introduction of Sequential Circuits](#)

**Combinational circuits** are defined as the time independent circuits which do not depend upon previous inputs to generate any output are termed as combinational circuits. **Sequential circuits** are those which are dependent on clock cycles and depend on present as well as past inputs to generate any output.

### Combinational Circuit –

1. In this output depends only upon present input.
2. Speed is fast.
3. It is designed easy.
4. There is no feedback between input and output.
5. This is time independent.
6. Elementary building blocks: Logic gates
7. Used for arithmetic as well as boolean operations.
8. Combinational circuits don't have capability to store any state.
9. As combinational circuits don't have clock, they don't require triggering.
10. These circuits do not have any memory element.
11. It is easy to use and handle.

**Examples** – Encoder, Decoder, Multiplexer, Demultiplexer

### Block Diagram –



Figure: Combinational Circuits

### Sequential Circuit –

1. In this output depends upon present as well as past input.
2. Speed is slow.
3. It is designed tough as compared to combinational circuits.
4. There exists a feedback path between input and output.
5. This is time dependent.
6. Elementary building blocks: Flip-flops
7. Mainly used for storing data.
8. Sequential circuits have capability to store any state or to retain earlier state.
9. As sequential circuits are clock dependent they need triggering.
10. These circuits have memory element.
11. It is not easy to use and handle.

**Examples** – Flip-flops, Counters

### Block Diagram –

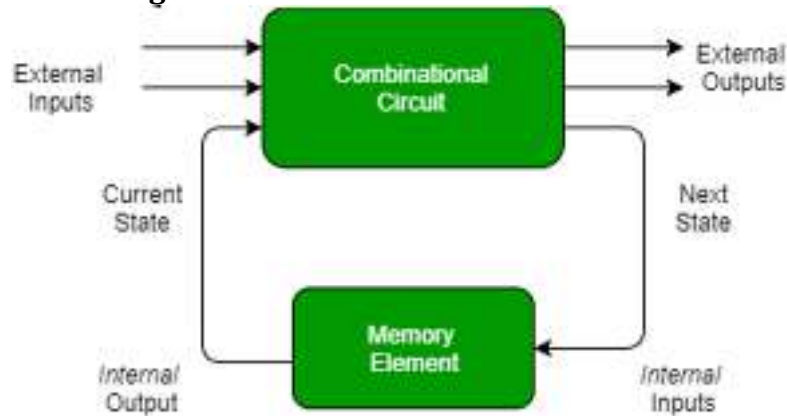
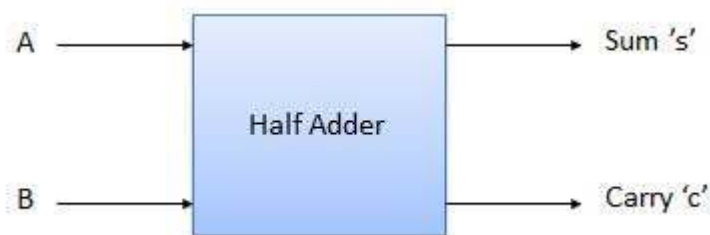


Figure: Sequential Circuit

## Half Adder

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 number A and B. It is the basic building block for addition of two **single** bit numbers. This circuit has two outputs **carry** and **sum**.

### Block diagram



### Truth Table

Inputs		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

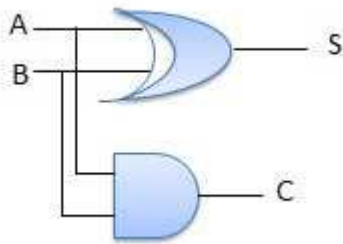
### Logical Expression:

$$\text{Sum} = A \text{ XOR } B$$

$$\text{Carry} = A \text{ AND } B$$

## Implementation:

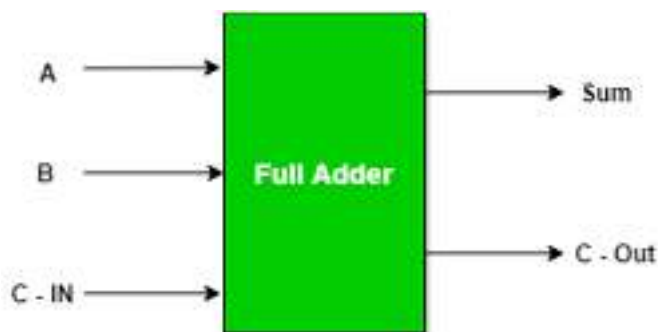
### Circuit Diagram



## Full Adder in Digital Logic

Full Adder is the adder which adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.

A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the another.



### Full Adder Truth Table:

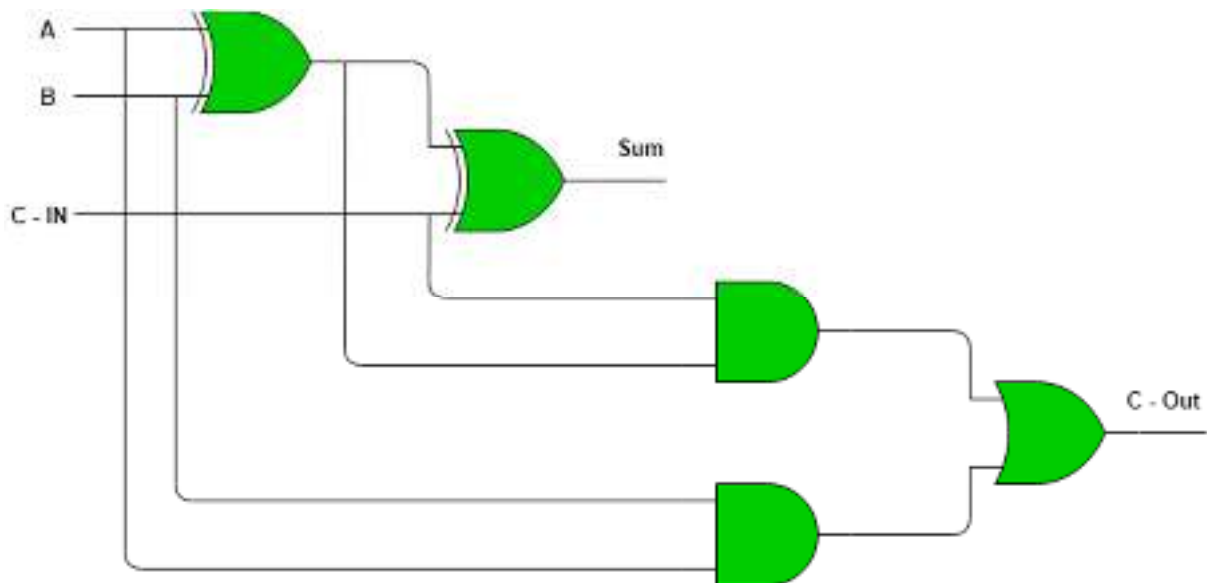
Inputs			Outputs	
A	B	C - IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

**Logical Expression for SUM:**

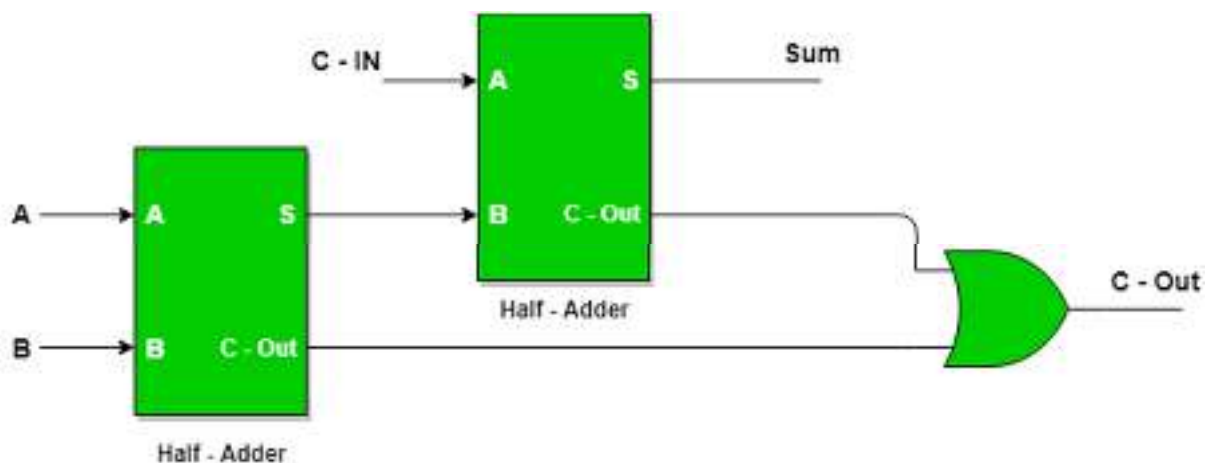
$$\begin{aligned} &= A' B' C\text{-IN} + A' B C\text{-IN}' + A B' C\text{-IN}' + A B C\text{-IN} \\ &= C\text{-IN} (A' B' + A B) + C\text{-IN}' (A' B + A B') \\ &= C\text{-IN} \text{ XOR } (A \text{ XOR } B) \\ &= (1,2,4,7) \end{aligned}$$

**Logical Expression for C-OUT:**

$$\begin{aligned} &= A' B C\text{-IN} + A B' C\text{-IN} + A B C\text{-IN}' + A B C\text{-IN} \\ &= (3,5,6,7) \end{aligned}$$

**Implementation of Full Adder using Half Adders**

2 Half Adders and a OR gate is required to implement a Full Adder.



With this logic circuit, two bits can be added together, taking a carry from the next lower order of magnitude, and sending a carry to the next higher order of magnitude.

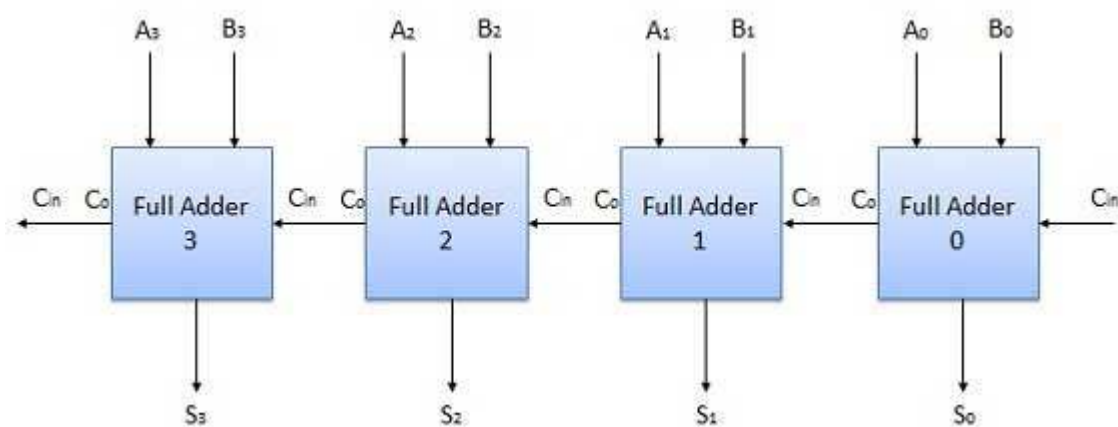
## N-Bit Parallel Adder

The Full Adder is capable of adding only two single digit binary number along with a carry input. But in practical we need to add binary numbers which are much longer than just one bit. To add two n-bit binary numbers we need to use the n-bit parallel adder. It uses a number of full adders in cascade. The carry output of the previous full adder is connected to carry input of the next full adder.

### 4 Bit Parallel Adder

In the block diagram,  $A_0$  and  $B_0$  represent the LSB of the four bit words A and B. Hence Full Adder-0 is the lowest stage. Hence its  $C_{in}$  has been permanently made 0. The rest of the connections are exactly same as those of n-bit parallel adder is shown in fig. The four bit parallel adder is a very common logic circuit.

### Block diagram



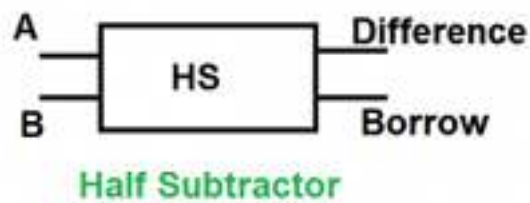
Eg:

$$\begin{array}{r} 1\ 0\ 1\ 1\ + \\ 1\ 0\ 0\ 1 \\ \hline 1\ 0\ 1\ 0\ 0 \end{array}$$

## Half Subtractor in Digital Logic(\*\*not included in syllabus)

Last Updated: 25-11-2019

Half Subtractor ( HS )



Truth Table

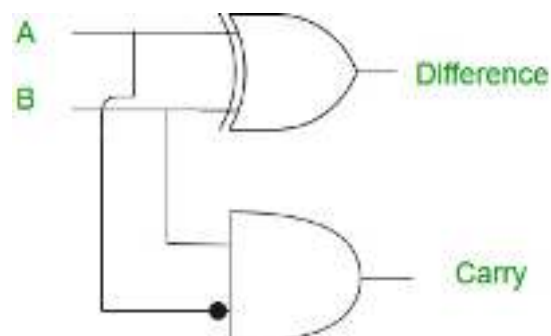
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Logical Expression

Difference =  $A \oplus B$

Borrow =  $A'B$

Implementation:



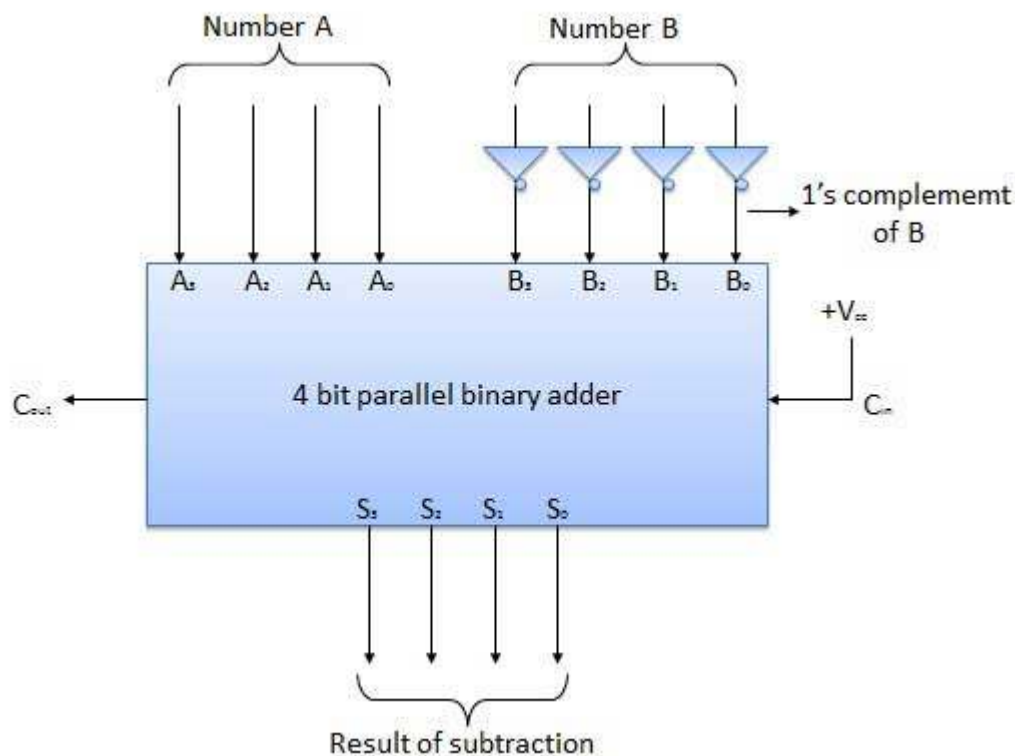
## N-Bit Parallel Subtractor

The subtraction can be carried out by taking the 1's or 2's complement of the number to be subtracted. For example we can perform the subtraction (A-B) by adding either 1's or 2's complement of B to A. That means we can use a binary adder to perform the binary subtraction.

## 4 Bit Parallel Subtractor

The number to be subtracted (B) is first passed through inverters to obtain its 1's complement. The 4-bit adder then adds A and 2's complement of B to produce the subtraction.  $S_3 S_2 S_1 S_0$  represents the result of binary subtraction (A-B) and carry output  $C_{out}$  represents the polarity of the result. If  $A > B$  then  $C_{out} = 0$  and the result of binary form (A-B) then  $C_{out} = 1$  and the result is in the 2's complement form.

### Block diagram



## Decoders –

Binary code of N digits can be used to store  $2^N$  distinct elements of coded information. This is what encoders and decoders are used for. **Encoders** convert  $2^N$  lines of input into a code of N bits and **Decoders** decode the N bits into  $2^N$  lines.

A decoding circuit is used to change the binary output of register was decoded into decimal form.

A decoder does the opposite job of an encoder. It is a combinational circuit that converts n lines of input into  $2^n$  lines of output. Let's take an example of 3-to-8 line decoder.

### Truth Table –

X	Y	Z	D0	D1	D2	D3	D4	D5	D6	D7
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

### Implementation –

D0 is high when  $X = 0$ ,  $Y = 0$  and  $Z = 0$ . Hence,  
 $D0 = X' Y' Z'$

Similarly,

$$D1 = X' Y' Z$$

$$D2 = X' Y Z'$$

$$D3 = X' Y Z$$

$$D4 = X Y' Z'$$

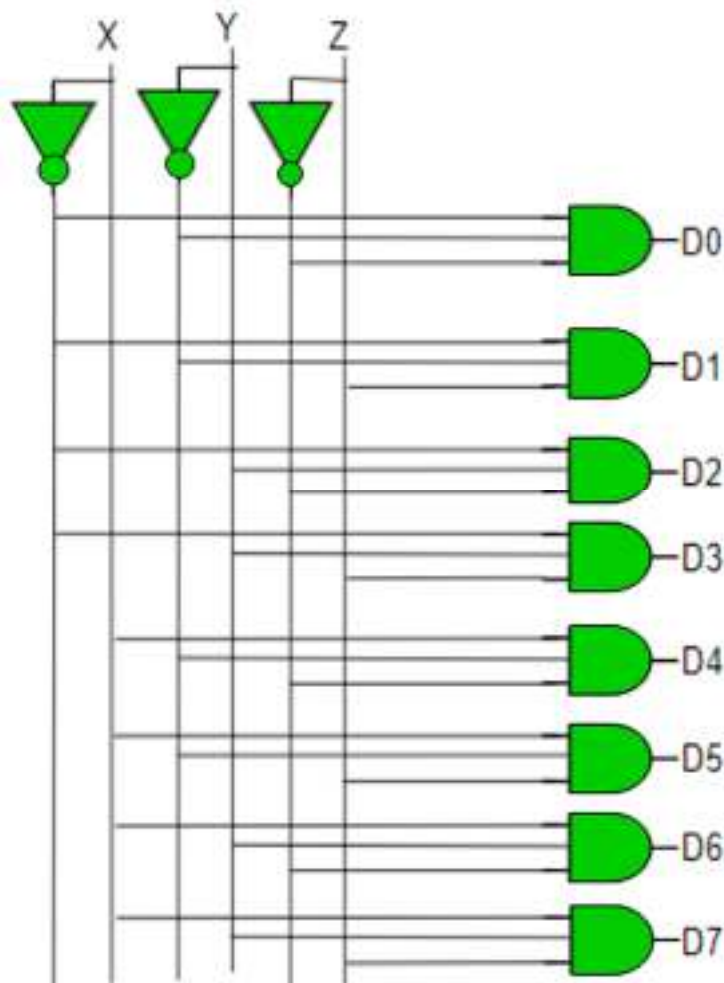
$$D5 = X Y' Z$$

$$D6 = X Y Z'$$

$$D7 = X Y Z$$

Hence,



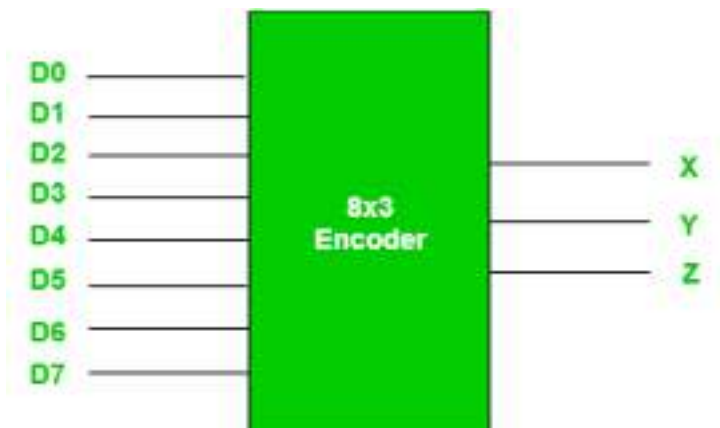


Same way we can construct 2 bit and 4 bit decoders based on our requirement .

### Encoders –

An encoder is a combinational circuit that converts binary information in the form of a  $2^N$  input lines into N output lines, which represent N bit code for the input. For simple encoders, it is assumed that only one input line is active at a time.

As an example, let's consider **Octal to Binary** encoder. As shown in the following figure, an octal-to-binary encoder takes 8 input lines and generates 3 output lines.



**Truth Table –**

D7	D6	D5	D4	D3	D2	D1	D0	X	Y	Z
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

As seen from the truth table, the output is 000 when D0 is active; 001 when D1 is active; 010 when D2 is active and so on.

### **Implementation –**

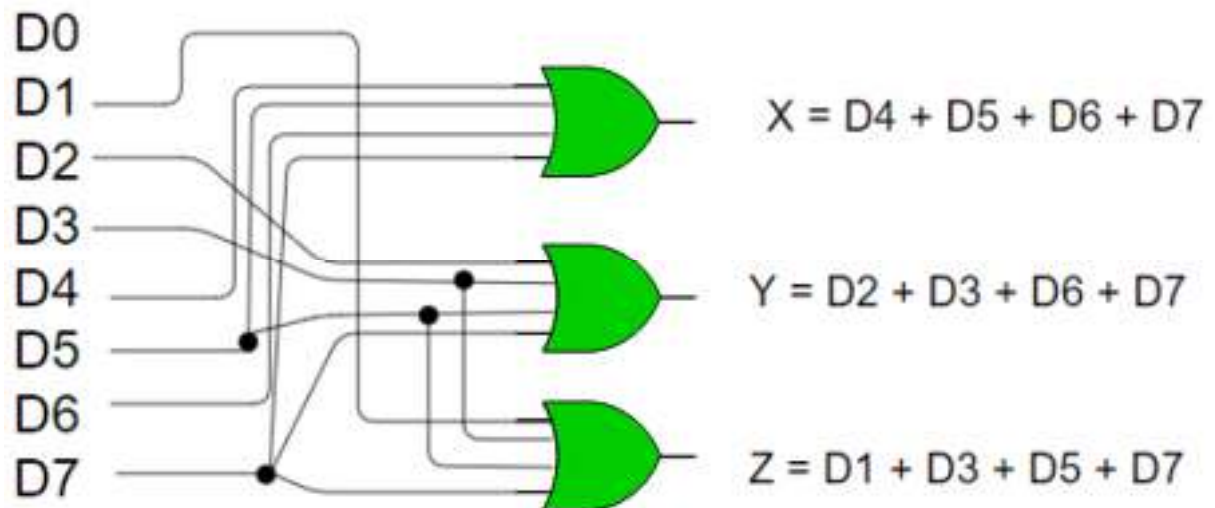
From the truth table, the output line Z is active when the input octal digit is 1, 3, 5 or 7. Similarly, Y is 1 when input octal digit is 2, 3, 6 or 7 and X is 1 for input octal digits 4, 5, 6 or 7. Hence, the Boolean functions would be:

$$X = D4 + D5 + D6 + D7$$

$$Y = D2 + D3 + D6 + D7$$

$$Z = D1 + D3 + D5 + D7$$

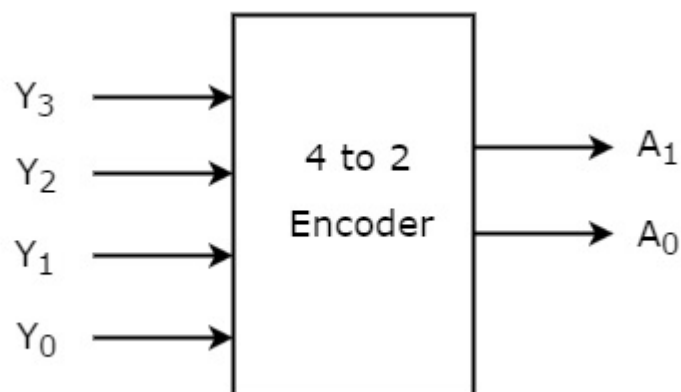
Hence, the encoder can be realised with OR gates as follows:



One limitation of this encoder is that only one input can be active at any given time.

## 4 to 2 Encoder

Let 4 to 2 Encoder has four inputs  $Y_3, Y_2, Y_1$  &  $Y_0$  and two outputs  $A_1$  &  $A_0$ . The **block diagram** of 4 to 2 Encoder is shown in the following figure.



At any time, only one of these 4 inputs can be '1' in order to get the respective binary code at the output. The **Truth table** of 4 to 2 encoder is shown below.

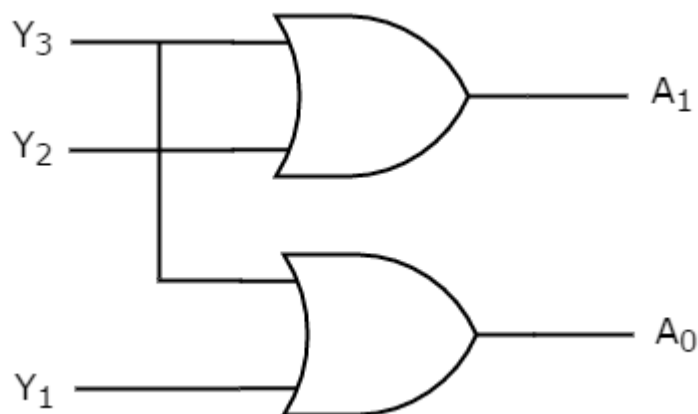
Inputs				Outputs	
$Y_3$	$Y_2$	$Y_1$	$Y_0$	$A_1$	$A_0$
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

From Truth table, we can write the **Boolean functions** for each output as

$$A_1 = Y_3 + Y_2$$

$$A_0 = Y_3 + Y_1$$

We can implement the above two Boolean functions by using two input OR gates. The **circuit diagram** of 4 to 2 encoder is shown in the following figure.



The above circuit diagram contains two OR gates. These OR gates encode the four inputs with two bits

## Applications

he encoders and decoders play an essential role in digital [electronics projects](#); encoders & decoders are used to convert data from one form to another form. These are frequently used in communication system such as telecommunication, networking, etc..to transfer data from one end to the other end. Similarly, in the digital domain, for easy transmission of data, it is often encrypted or placed within codes, and then transmitted. At the receiver, the coded data

is decrypted or gathered from the code and is processed in order to be displayed or given to the load accordingly.

## Multiplexer:( It is known as **Data Selector**)

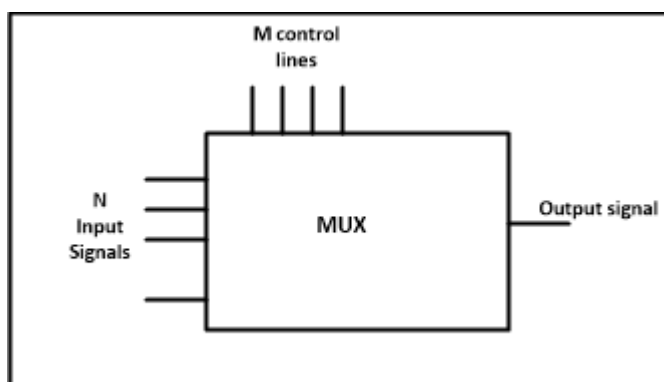
Multiplexer means many into one. A multiplexer is a circuit used to select and route any one of the several input signals to a signal output. An simple example of an non electronic circuit of a multiplexer is a single pole multiposition switch.

Multiposition switches are widely used in many [electronics circuits](#). However circuits that operate at high speed require the multiplexer to be automatically selected. A mechanical switch cannot perform this task satisfactorily. Therefore, multiplexer used to perform high speed switching are constructed of electronic components.

Multiplexer handle two type of data that is analog and digital. For analog application, multiplexer are built of relays and transistor switches. For digital application, they are built from standard logic gates.

The multiplexer used for digital applications, also called digital multiplexer, is a circuit with many input but only one output. By applying control signals, we can steer any input to the output. Few types of multiplexer are 2-to-1, 4-to-1, 8-to-1, 16-to-1 multiplexer.

Following figure shows the general idea of a multiplexer with n input signal, m control signals and one output signal.

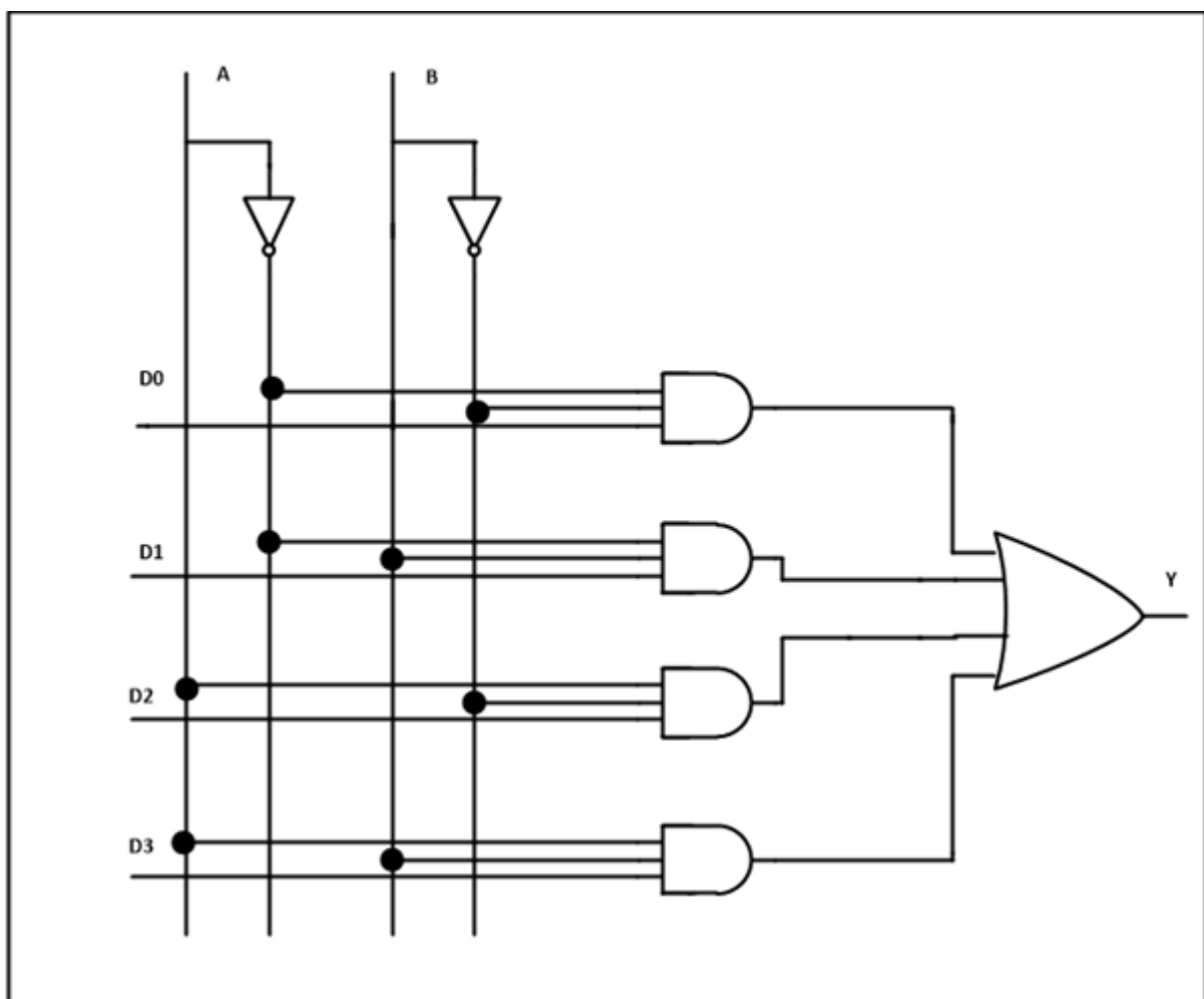


**Multiplexer Pin Diagram**

## Understanding 4-to-1 Multiplexer:

The 4-to-1 multiplexer has 4 input bit, 2 control bits, and 1 output bit. The four input bits are D0,D1,D2 and D3. only one of this is transmitted to the output y. The output depends on the value of AB which is the control input. The control input determines which of the input data bit is transmitted to the output.

For instance, as shown in fig. when  $AB = 00$ , the upper AND gate is enabled while all other AND gates are disabled. Therefore, data bit D0 is transmitted to the output, giving  $Y = D_0$ .



4 to 1 Multiplexer Circuit Diagram – [ElectronicsHub.Org](https://www.electronicshub.org)

If the control input is changed to  $AB = 11$ , all gates are disabled except the bottom AND gate. In this case, D3 is transmitted to the output and  $Y = D_3$ .

- An example of 4-to-1 multiplexer is IC 74153 in which the output is same as the input.

- Another example of 4-to-1 multiplexer is 45352 in which the output is the complement of the input.
- Example of 16-to-1 line multiplexer is IC74150.

### *Applications of Multiplexer:*

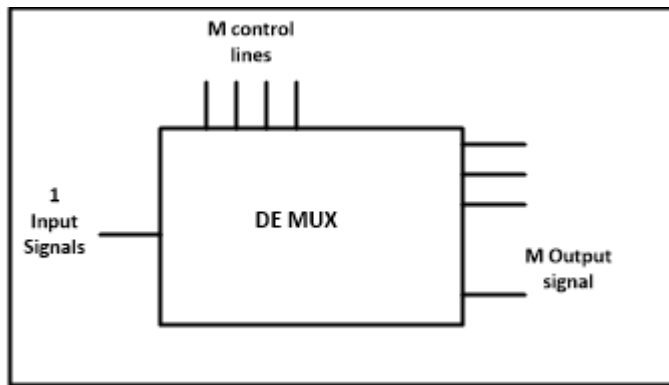
Multiplexer are used in various fields where multiple data need to be transmitted using a single line. Following are some of the applications of multiplexers –

1. **Communication system** – Communication system is a set of system that enable communication like transmission system, relay and tributary station, and communication network. The efficiency of communication system can be increased considerably using multiplexer. Multiplexer allow the process of transmitting different type of data such as audio, video at the same time using a single transmission line.
2. **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.
3. **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.
4. **Transmission from the computer system of a satellite** – Multiplexer can be used for the transmission of data signals from the computer system of a satellite or spacecraft to the ground system using the GPS (Global Positioning System) satellites.

### **Demultiplexer:**

Demultiplexer means one to many. A demultiplexer is a circuit with one input and many output. By applying control signal, we can steer any input to the output. Few types of demultiplexer are 1-to 2, 1-to-4, 1-to-8 and 1-to 16 demultiplexer.

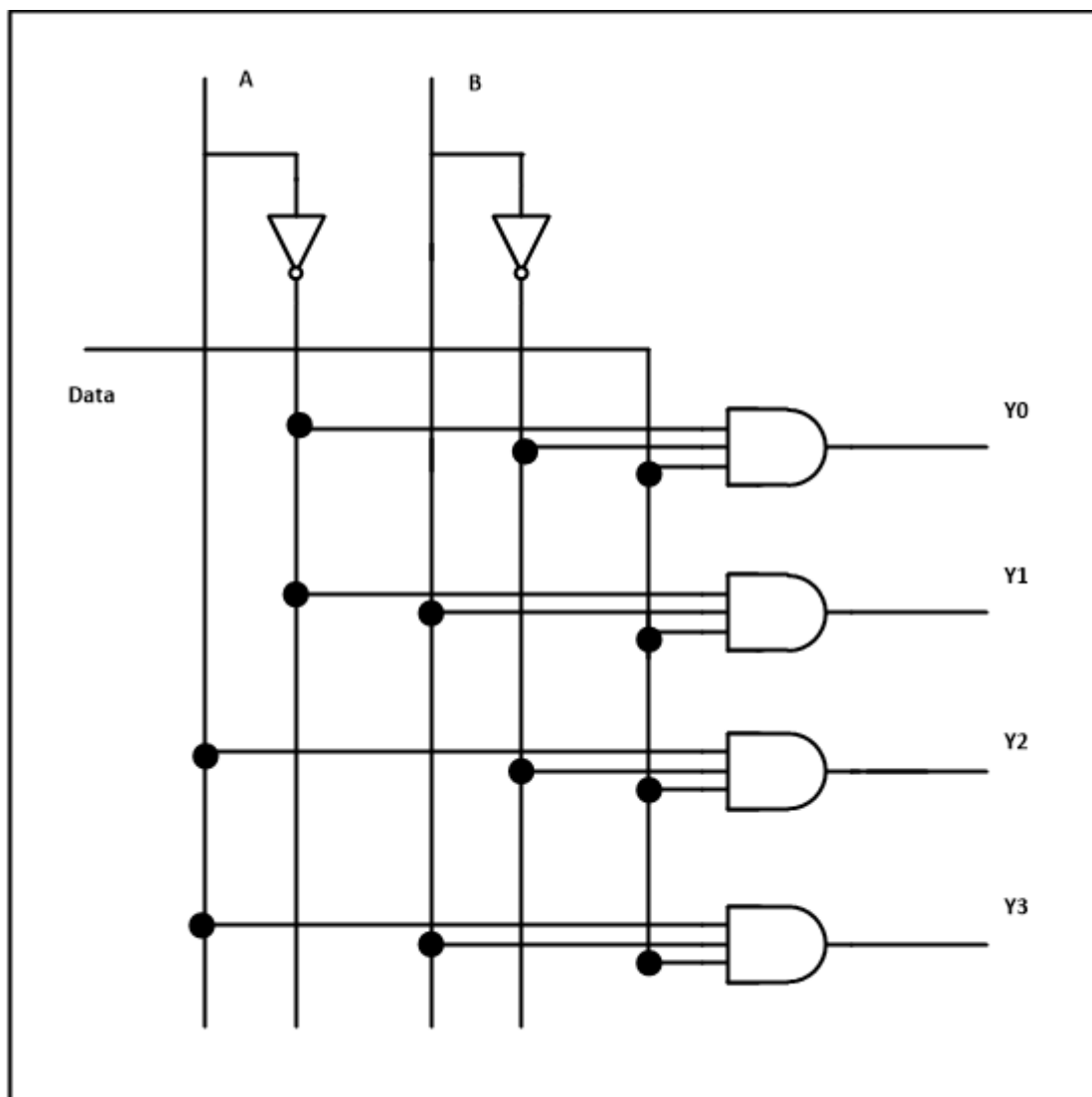
Following figure illustrate the general idea of a demultiplexer with 1 input signal, m control signals, and n output signals.



Demultiplexer Pin Diagram

## Understanding 1- to-4 Demultiplexer:

The 1-to-4 demultiplexer has 1 input bit, 2 control bit, and 4 output bits. An example of 1-to-4 demultiplexer is IC 74155. The 1-to-4 demultiplexer is shown in figure below-





#### 1 to 4 Demultiplexer Circuit Diagram – [ElectronicsHub.Org](https://www.electronicshub.org)

The input bit is labelled as Data D. This data bit is transmitted to the data bit of the output lines. This depends on the value of AB, the control input.

When  $AB = 01$ , the upper second AND gate is enabled while other AND gates are disabled. Therefore, only data bit D is transmitted to the output, giving  $Y1 = \text{Data}$ .

If D is low, Y1 is low. If D is high, Y1 is high. The value of Y1 depends upon the value of D. All other outputs are in low state.

If the control input is changed to  $AB = 10$ , all the gates are disabled except the third AND gate from the top. Then, D is transmitted only to the Y2 output, and  $Y2 = \text{Data}$ .

Example of 1-to-16 demultiplexer is IC 74154 it has 1 input bit, 4 control bits and 16 output bit.

#### *Applications of Demultiplexer:*

1. Demultiplexer is used to connect a single source to multiple destinations. The main application area of demultiplexer is communication system where multiplexer are used. Most of the communication system are bidirectional i.e. they function in both ways (transmitting and receiving signals). Hence, for most of the applications, the multiplexer and demultiplexer work in sync. Demultiplexer are also used for reconstruction of parallel data and ALU circuits.
2. **Communication System** – Communication system use multiplexer to carry multiple data like audio, video and other form of data using a single line for transmission. This process make the transmission easier. The demultiplexer receive the output signals of the multiplexer and converts them back to the original form of the data at the receiving end. The multiplexer and demultiplexer work together to carry out the process of transmission and reception of data in communication system.
3. **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 demultiplexer. The output of ALU is fed as the data input to the demultiplexer. Each output of demultiplexer is connected to multiple register which can be stored in the registers.
4. **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 from the incoming serial data stream is given as

data input to the demultiplexer at the regular intervals. A counter is attach to the control input of the demultiplexer. This counter directs the data signal to the output of the demultiplexer where these data signals are stored. When all data signals have been stored, the output of the demultiplexer can be retrieved and read out in parallel.