

MODULE 2

Combinational Logic:

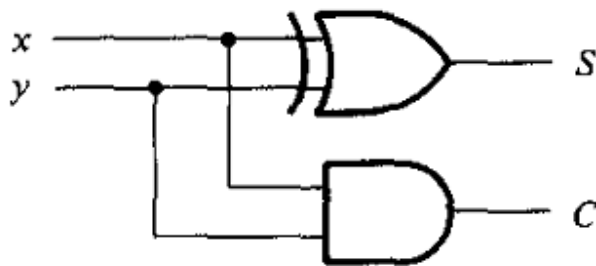
Combinational Circuits

Circuits in which all outputs at any given time depend only on the inputs at that time are called combinational logic circuits.

A combinational circuit performs a specific information-processing operation fully specified logically by a set of Boolean functions. Sequential circuits employ memory elements (binary cells) in addition to logic gates. Their outputs are a function of the inputs and the state of the memory elements. The state of memory elements, in turn, is a function of previous inputs. As a consequence, the outputs of a sequential circuit depend not only on present inputs, but also on past inputs, and the circuit behavior must be specified by a time sequence of inputs and internal states.

1 BINARY ADDERS

One of the most important tasks performed by a digital computer is the operation of adding two binary numbers.



$$(e) \begin{aligned} S &= x \oplus y \\ C &= xy \end{aligned}$$

Half adder circuit

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Truth table of half adder

The simplified Boolean functions for the two outputs can be obtained directly from the truth table. The simplified sum of products expressions are

$$S = x'y + xy'$$

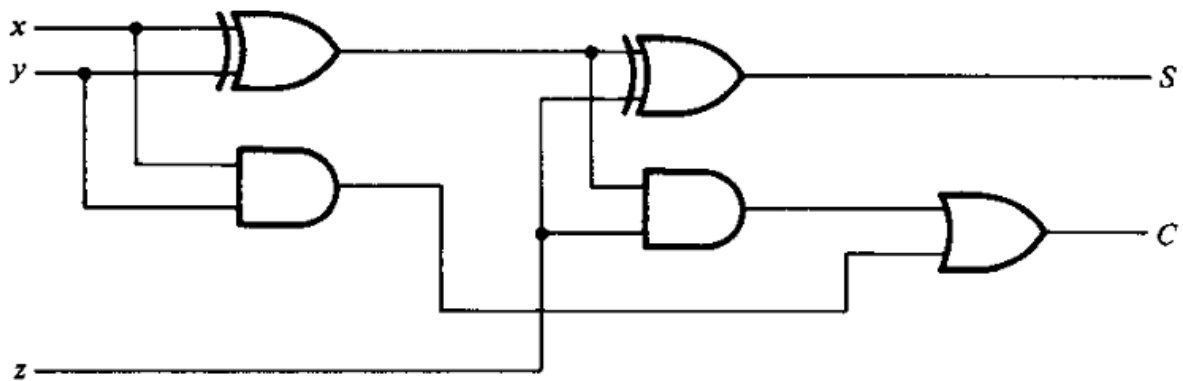
$$C = xy$$

Full-Adder

A full-adder is a combinational circuit that forms the arithmetic sum of three input bits. It consists of three inputs and two outputs.

Two of the input variables, denoted by x and y , represent the two significant bits to be added. The third input, z , represents the carry from the previous lower significant position. The two outputs are designated by the symbols S for sum and C for carry. The binary variable S gives the value of the least significant bit of the sum. The binary variable C gives the output carry. The truth table of the full-adder is

x	y	z	C	S
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



Implementation of full adder

This implementation uses the following Boolean expressions:

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

$$C = xy + xz + yz$$

Ripple-Carry Adder

The problem of adding two multidigit binary numbers has the following form.

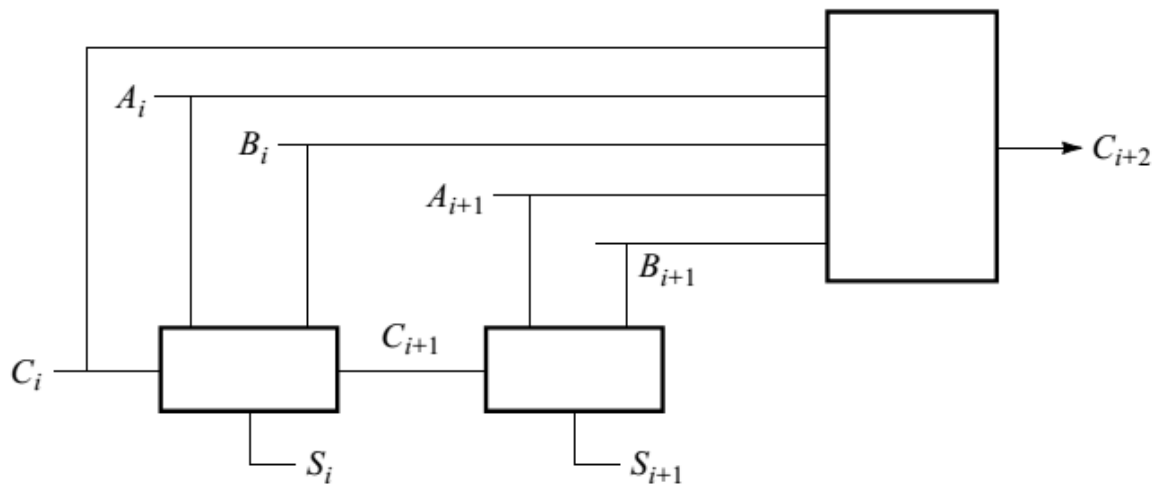
Two n -bit binary numbers are available, with all digits being presented in parallel. The addition is performed by using a full adder to add each corresponding pair of digits, one from each number. The full adders are connected in tandem so that the carry out from one stage becomes the carry into the next stage, as illustrated for the case of four-digit numbers in Figure 4. Thus, the carry ripples through each stage. For binary addition, the carry into the first (least significant) stage is 0. The last carry out (the overflowcarry) becomes the most significant bit of the $(n+1)$ -bit sum.

Since the carry of each full adder has a propagation delay of $2t_p$, the total delay in carrying out the sum of two n -bit numbers is $2nt_p$. Not every pair of two-bit numbers will experience this much delay. Take the following two numbers

as an example:

101010

010101



Carry-lookahead circuit schematic

Assuming that the carry into the first stage is zero, no carries are generated at any stage in taking the sum. Hence, there will be no carry ripple, and so no propagation delay along the carry chain.

However, to handle the general case, provision must be made for the worst case; no new numbers should be presented for addition before the total delay represented by the worst case. The maximum addition speed, thus, is limited by the worst case of carry propagation delay.

Carry-Lookahead Adder

In contemplating the addition of two n -digit binary numbers, we were appalled by the thought of a single combinational circuit with all those inputs. So we considered the repeated use of a simpler circuit, a full adder, with the least possible number of inputs. But what is gained in circuit simplicity with this approach is lost in speed. Since the speed is limited by the delay in the carry function, some of the lost speed might be regained if we could design a circuit—just for the carry—with more inputs than 2 but not as many as $2n$. Suppose that several full-adder stages are treated as a unit. The inputs to the unit are the carry into the unit as well as the input digits to all the full adders in that unit. Then perhaps the carry out could be obtained faster than the ripple carry through the same number of full adders.

These concepts are illustrated in above figure with a unit consisting of just two full adders and a carry-lookahead circuit. The four digits to be added, as well as the input carry C_i , are present simultaneously. It is possible to get an expression for the carry out, C_{i+2} , from the unit by using the expression for the carry of the full adder

For reasons which will become clear shortly, let's attach names to the two terms in the carry expression, changing the names of the variables to A and B from x and y in accordance with above figure.

Define the generated carry G_i and the propagated carry P_i for the i th full adder as follows:

$$G_i = A_i B_i$$

$$P_i = A_i \oplus B_i$$

Inserting these into the expression for the carryout gives

$$C_{i+1} = A_i B_i + C_i (A_i \oplus B_i) = G_i + P_i C_i$$

Subtractor

Subtractor circuits take two binary numbers as input and subtract one binary number input from the other binary number input. Similar to adders, it gives out two outputs, difference and borrow (carry-in the case of Adder). There are two types of subtractors.

- Half Subtractor
- Full Subtractor

Half Subtractor

The half-subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, X (minuend) and Y (subtrahend) and two outputs D (difference) and B (borrow). The logic symbol and truth table are shown below.

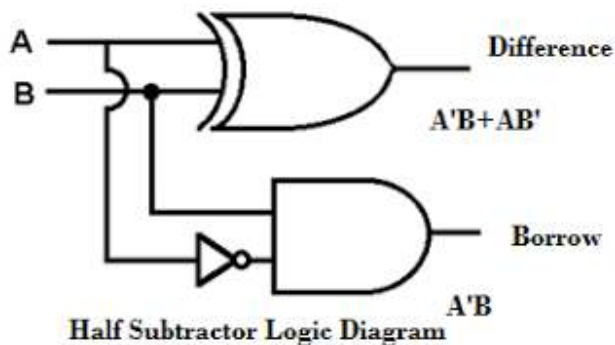
Truth Table

X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$\text{Difference} = A'B + AB' = A \oplus B$$

$$\text{Borrow} = A'B$$

The logic Diagram of Half Subtractor is shown below.



Full Subtractor

A full subtractor is a combinational circuit that performs subtraction involving three bits, namely minuend, subtrahend, and borrow-in. so it allows *cascading* which results in the possibility of **multi-bit subtraction**. The truth table for a full subtractor is given below.

Truth Table

X	Y	Bin	D	Bout
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

$$\text{Difference} = A'B'C + A'BB' + AB'C' + ABC$$

Reduce it like adder

Then We got

$$\text{Difference} = A \oplus B \oplus C$$

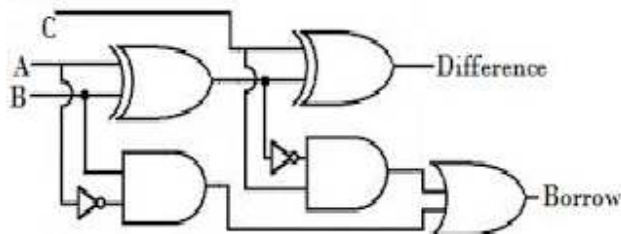
$$\text{Borrow} = A'B'C + A'BC' + A'BC + ABC$$

$$= A'B'C + A'BC' + A'BC + A'BC + A'BC + ABC \quad \text{----->} \quad A'BC = A'BC + A'BC + A'BC$$

$$=A'C(B'+B)+A'B(C'+C)+BC(A'+A)$$

$$\text{Borrow}=A'C+A'B+BC$$

The logic diagram of Full Subtractor is shown below



Full Subtractor-Logic Diagram

Multiplexers:-

Multiplexing means transmitting a large number of information units over a smaller number of channels or lines. A digital multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line.

The selection of a particular input line is controlled by a set of selection lines. Normally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.

A 4-to-1-line multiplexer is shown in given Fig. Each of the four input lines, I_0 to I_3 is applied to one input of an AND gate. Selection lines S_1 and S_0 are decoded to select a particular AND gate. The function table, shown in given , lists the input-to-output path for each possible bit combination of the selection lines. When this MSI function is used in the design of a digital system, it is represented in block diagram form, as shown in given Fig. To demonstrate the circuit operation, consider the case when $S_1S_0 = 10$. The AND gate associated with input I_2 , has two of its inputs equal to 1 and the third input connected to I_2 . The other three AND gates have at least one input equal to 0, which makes their outputs equal to 0. The OR gate output is now equal to the value of I_2 thus providing a path from the selected input to the output. A multiplexer is also called a data selector, since it selects one of many inputs and steers the binary information to the output line.

The AND gates and inverters in the multiplexer resemble a decoder circuit and, indeed, they decode the input-selection lines. In general, a 2^n -to-1-line multiplexer is constructed from an n -to- 2^n decoder by adding to it 2^n input lines, one to each AND gate. The outputs of the AND gates are applied to a single OR gate to provide the 1-line output. The size of a multiplexer is

specified by the number 2^n of its input lines and the single output line. It is then implied that it also contains n selection lines. A multiplexer is often abbreviated as MUX.

As in decoders, multiplexer les may have an enable input to control the operation of the unit. When the enable input is in a given binary state, the outputs are disabled, and when it is in the other state (the enable state), the circuit functions as a normal multiplexer. The enable input (sometimes called strobe) can be used to expand two or more multiplexerles to a digital multiplexer with a larger number of inputs.

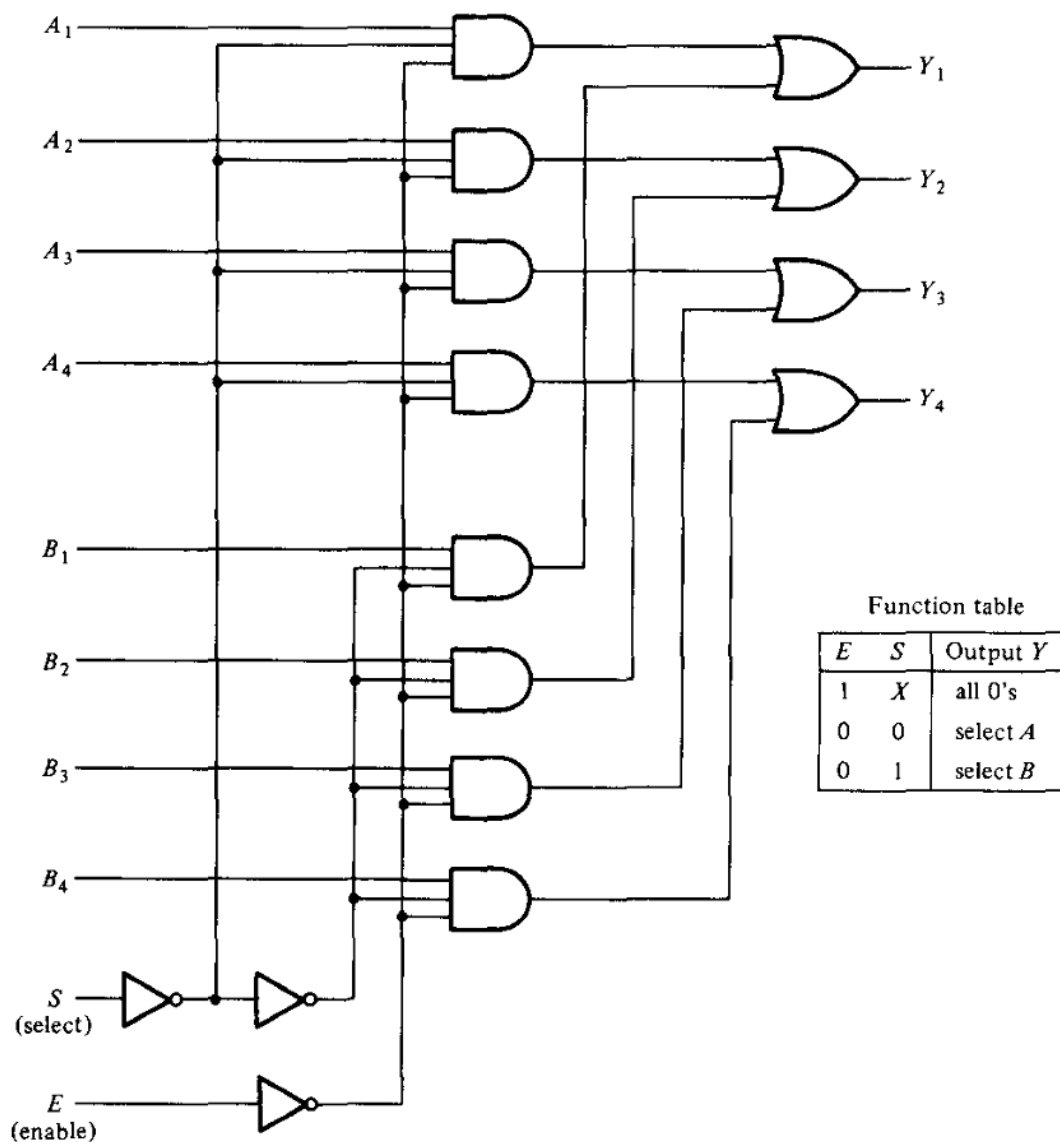


Fig 2 to 1 line multiplexer

As shown in the function table, the unit is selected when $E = 0$. Then, if $S = 0$, the four A inputs have a path to the outputs. On the other hand, if $S = 1$, the four B inputs are selected. The outputs have all D's when $E = 1$, regardless of the value of S.

Applications of Multiplexer:

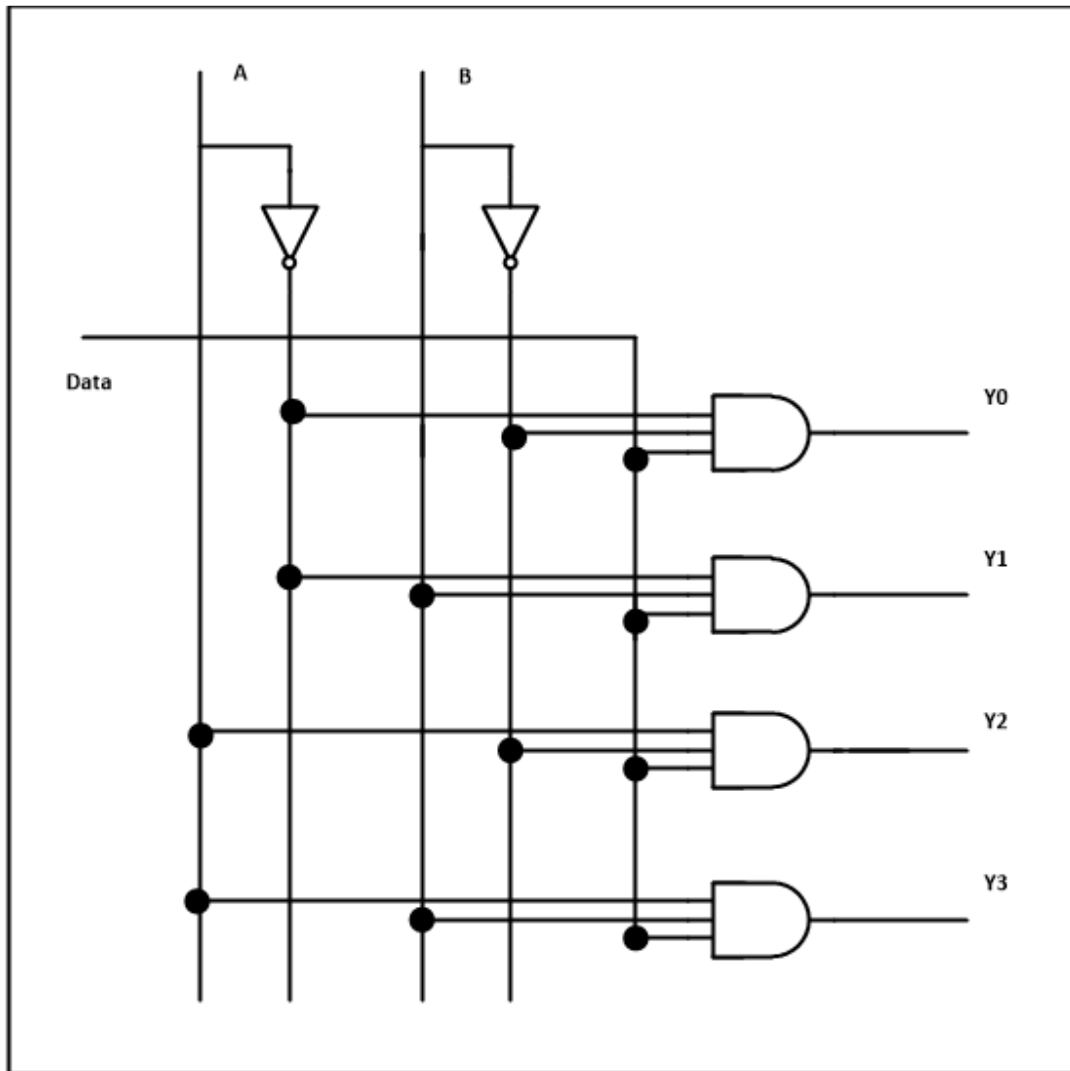
Multiplexer are used in various fields where multiple data need to be transmitted using a single line. Following are some of the applications of multiplexers -

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. **Computermemory** - 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.



1 to 4 Demultiplexer Circuit Diagram

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.

The Digital Comparator

Another common and very useful combinational logic circuit is that of the **Digital Comparator** circuit. Digital or Binary Comparators are made up from standard AND, NOR and NOT gates that compare the digital signals present at their input terminals and produce an output depending upon the condition of those inputs.

For example, along with being able to add and subtract binary numbers we need to be able to compare them and determine whether the value of input A is greater than, smaller than or equal to the value at input B etc. The digital comparator accomplishes this using several logic gates that operate on the principles of Boolean Algebra. There are two main types of **Digital Comparator** available and these are.

- 1. Identity Comparator – an *Identity Comparator* is a digital comparator that has only one output terminal for when $A = B$ either “HIGH” $A = B = 1$ or “LOW” $A = B = 0$
- 2. Magnitude Comparator – a *Magnitude Comparator* is a type of digital comparator that has three output terminals, one each for equality, $A = B$ greater than, $A > B$ and less than $A < B$

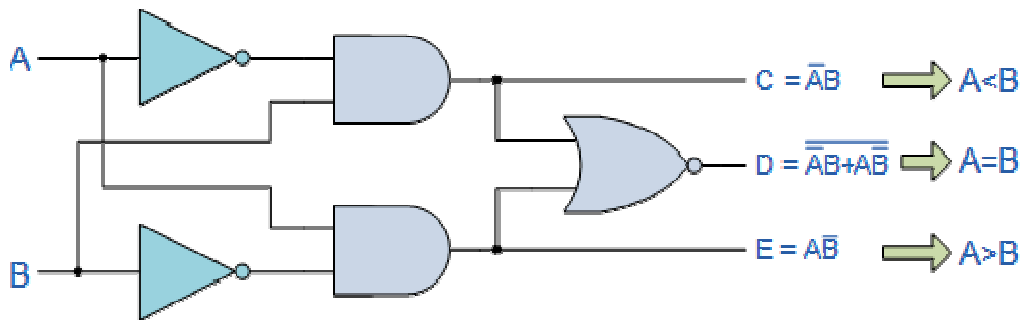
The purpose of a **Digital Comparator** is to compare a set of variables or unknown numbers, for example A (A1, A2, A3, An, etc) against that of a constant or unknown value such as B (B1, B2, B3,Bn, etc) and produce an output condition or flag depending upon the result of the comparison. For example, a magnitude comparator of two 1-bits, (A and B) inputs would produce the following three output conditions when compared to each other.

$$A > B, \quad A = B, \quad A < B$$

Which means: A is greater than B, A is equal to B, and A is less than B

This is useful if we want to compare two variables and want to produce an output when any of the above three conditions are achieved. For example, produce an output from a counter when a certain count number is reached. Consider the simple 1-bit comparator below.

1-bit Digital Comparator



Then the operation of a 1-bit digital comparator is given in the following Truth Table.

Digital Comparator Truth Table

Inputs		Outputs		
B	A	A > B	A = B	A < B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

You may notice two distinct features about the comparator from the above truth table. Firstly, the circuit does not distinguish between either two “0” or two “1”s as an output $A = B$ is produced when they are both equal, either $A = B = “0”$ or $A = B = “1”$. Secondly, the output condition for $A = B$ resembles that of a commonly available logic gate, the Exclusive-NOR or Ex-NOR function (equivalence) on each of the n-bits giving: $Q = A \oplus B$

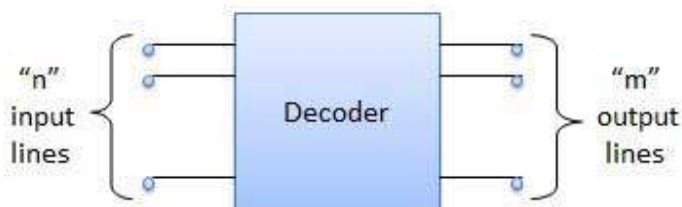
Digital comparators actually use Exclusive-NOR gates within their design for comparing their respective pairs of bits. When we are comparing two binary or BCD values or variables against each other, we are comparing the “magnitude” of these values, a logic “0” against a logic “1” which is where the term **Magnitude Comparator** comes from.

As well as comparing individual bits, we can design larger bit comparators by cascading together n of these and produce an n-bit comparator just as we did for the n-bit adder in the previous tutorial. Multi-bit comparators can be constructed to compare whole binary or BCD words to produce an output if one word is larger, equal to or less than the other.

Decoder

A decoder is a combinational circuit. It has n input and to a maximum $m = 2^n$ outputs. Decoder is identical to a demultiplexer without any data input. It performs operations which are exactly opposite to those of an encoder.

Block diagram



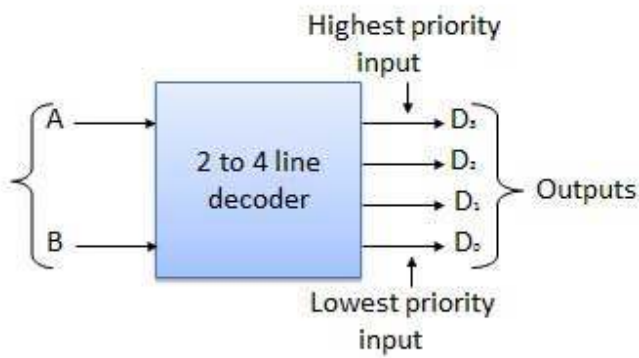
Examples of Decoders are following.

- Code converters
- BCD to seven segment decoders
- Nixie tube decoders
- Relay actuator

2 to 4 Line Decoder

The block diagram of 2 to 4 line decoder is shown in the fig. A and B are the two inputs where D through D are the four outputs. Truth table explains the operations of a decoder. It shows that each output is 1 for only a specific combination of inputs.

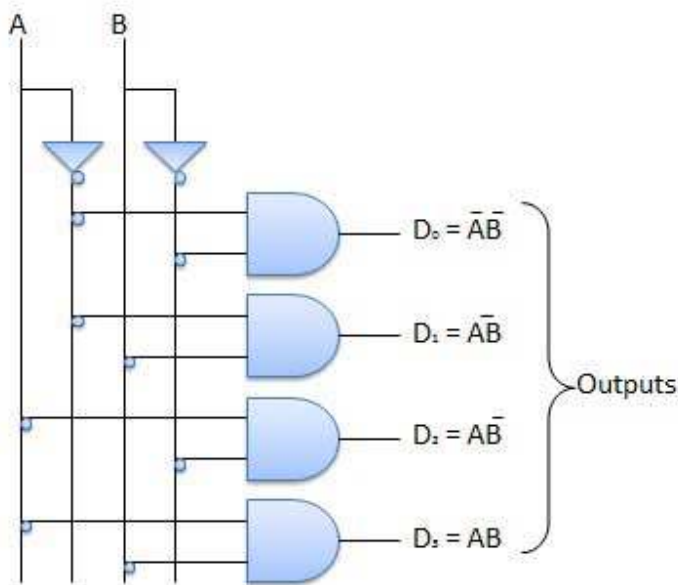
Block diagram



Truth Table

Inputs		Output			
A	B	D ₀	D ₁	D ₂	D ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

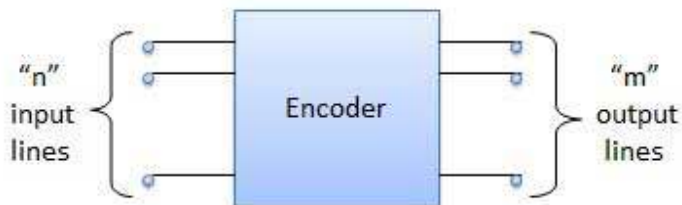
Logic Circuit



Encoder

Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder. An encoder has n number of input lines and m number of output lines. An encoder produces an m bit binary code corresponding to the digital input number. The encoder accepts an n input digital word and converts it into an m bit another digital word.

Block diagram



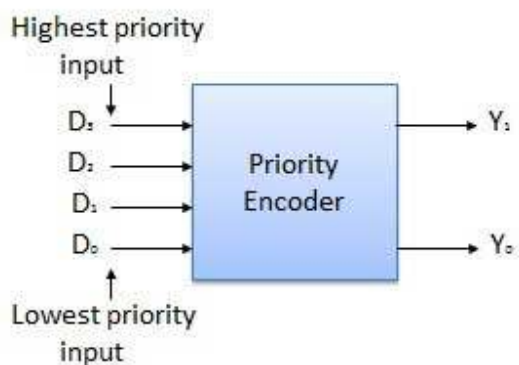
Examples of Encoders are following.

- Priority encoders
- Decimal to BCD encoder
- Octal to binary encoder
- Hexadecimal to binary encoder

Priority Encoder

This is a special type of encoder. Priority is given to the input lines. If two or more input line are 1 at the same time, then the input line with highest priority will be considered. There are four input D_0, D_1, D_2, D_3 and two output Y_0, Y_1 . Out of the four input D_3 has the highest priority and D_0 has the lowest priority. That means if $D_3 = 1$ then $Y_1 Y_0 = 11$ irrespective of the other inputs. Similarly if $D_3 = 0$ and $D_2 = 1$ then $Y_1 Y_0 = 10$ irrespective of the other inputs.

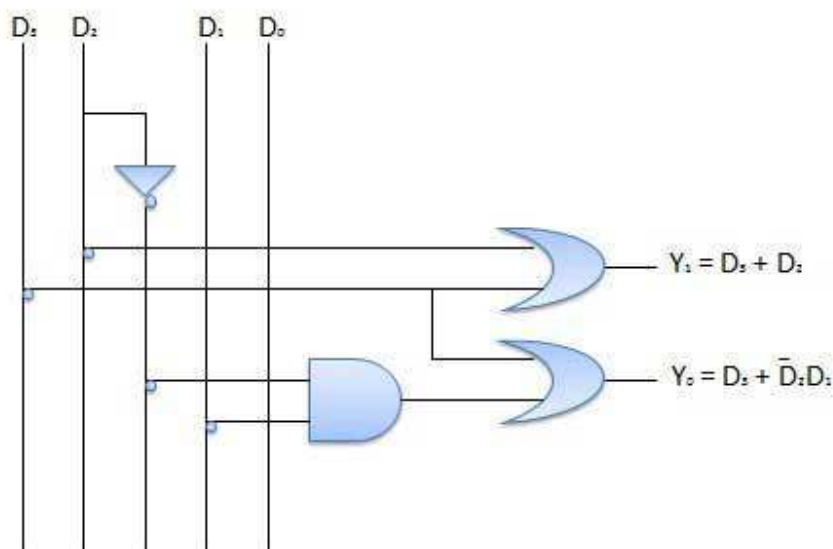
Block diagram



Truth Table

Highest	Inputs		Lowest	Outputs	
D_2	D_1	D_0	D_0	Y_0	Y_1
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1

Logic Circuit



PARITY GENERATOR AND CHECKER

- Parity is a very useful tool in information processing in digital computers to indicate any presence of error in bit information.
- External noise and loss of signal strength cause loss of data bit information while transporting data from one device to other device, located inside the computer or externally.

•To indicate any occurrence of error, an extra bit is included with the message according to the total number of 1s in a set of data, which is called parity.

•If the extra bit is considered 0 if the total number of 1s is even and 1 for odd quantities of 1s in a set of data, then it is called even parity.

•On the other hand, if the extra bit is 1 for even quantities of 1s and 0 for an odd number of 1s, then it is called odd parity

Parity Generator:-

A parity generator is a combination logic system to generate the parity bit at the transmitting side

<i>Four bit Message</i> $D_3D_2D_1D_0$	<i>Even Parity</i> (P_e)	<i>Odd Parity</i> (P_o)
0000	0	1
0001	1	0
0010	1	0
0011	0	1
0100	1	0
0101	0	1
0110	0	1
0111	1	0
1000	1	0
1001	0	1
1010	0	1
1011	1	0
1100	0	1
1101	1	0
1110	1	0
1111	0	1

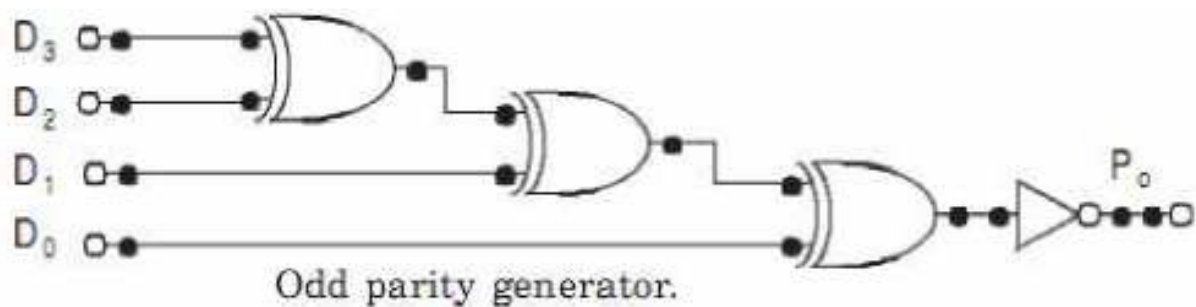
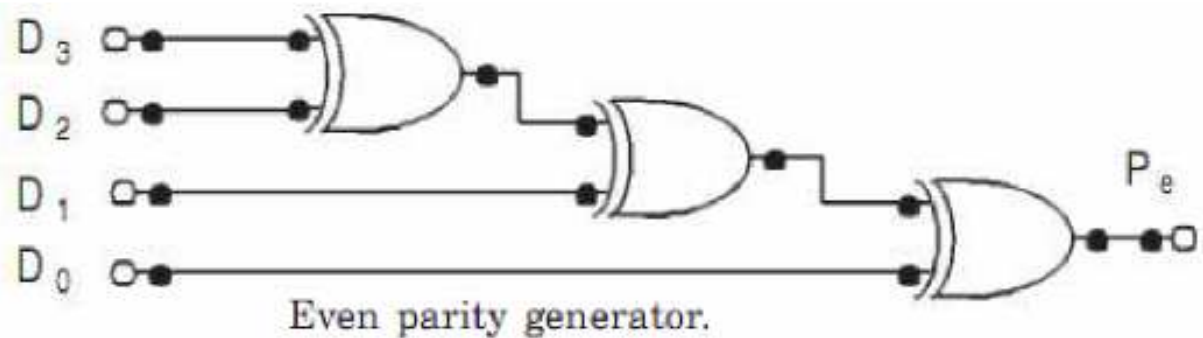
If the message bit combination is designated as $D_3D_2D_1D_0$, and P_e, P_o are the even and odd parity respectively, then it is obvious from the table that the Boolean expressions of even

parity and odd parity are

$$P_e = D_3 \oplus D_2 \oplus D_1 \oplus D_0$$

and

$$P_o = (D_3 \oplus D_2 \oplus D_1 \oplus D_0)'$$

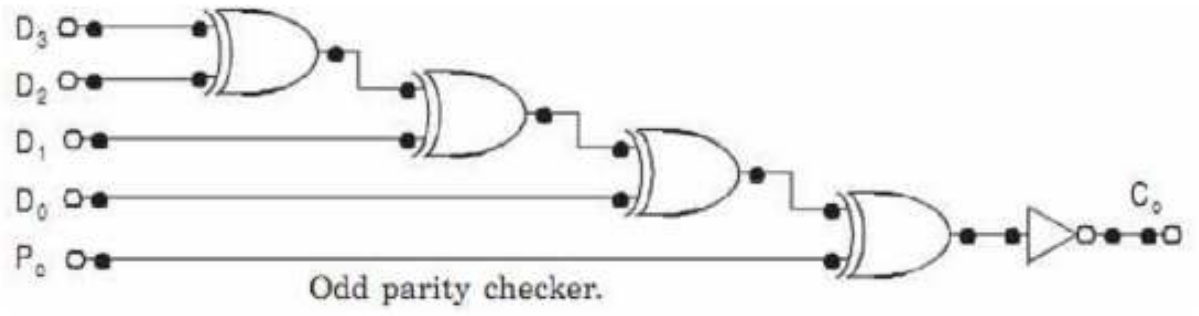
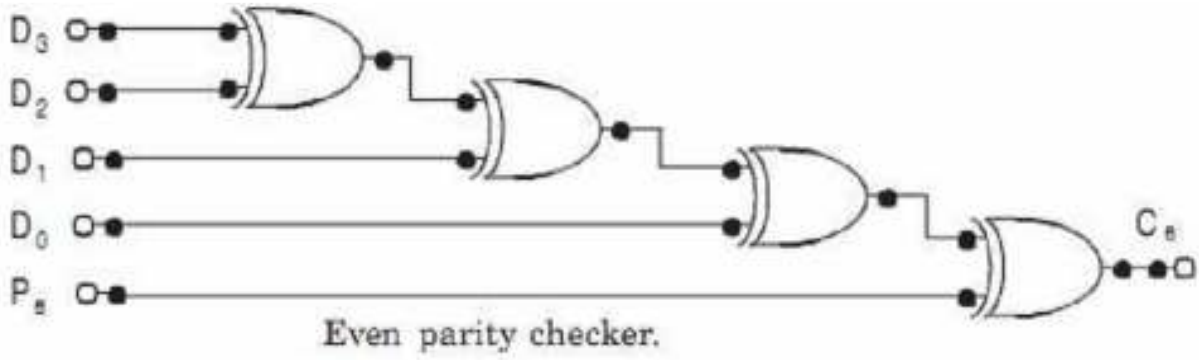


The above illustration is given for a message with four bits of information. However, the logic diagrams can be expanded with more XOR gates for any number of bits.

Parity Checker

- The message bits with the parity bit are transmitted to their destination, where they are applied to a parity checker circuit.
- The circuit that checks the parity at the receiver side is called the parity checker. The parity checker circuit produces a check bit and is very similar to the parity generator circuit.
- If the check bit is 1, then it is assumed that the received data is incorrect. The check bit will be 0 if the received data is correct.

Note that the check bit is 0 for all the bit combinations of correct data. For incorrect data the parity check bit will be another logic value



4-bit message $D_3D_2D_1D_0$	Even Parity (P_e)	Even Parity Checker (C_e)
0000	0	0
0001	1	0
0010	1	0
0011	0	0
0100	1	0
0101	0	0
0110	0	0
0111	1	0
1000	1	0
1001	0	0
1010	0	0
1011	1	0
1100	0	0
1101	1	0
1110	1	0
1111	0	0

Even parity checker.

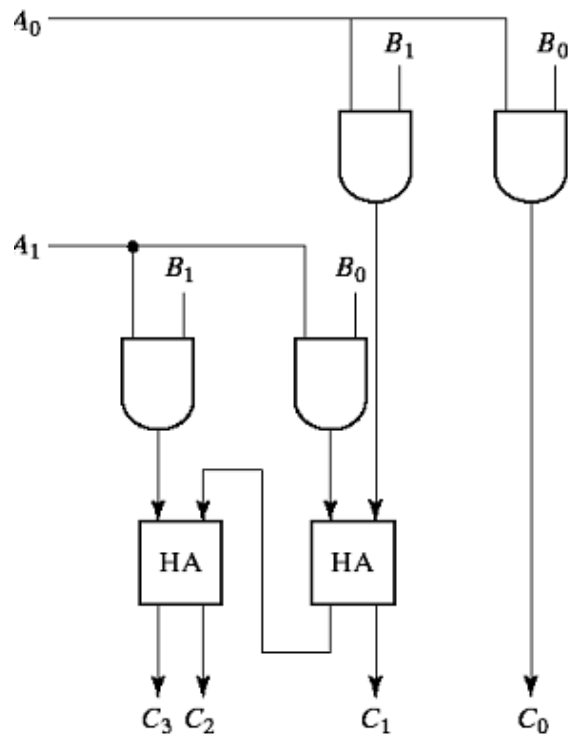
4-bit message $D_3D_2D_1D_0$	Odd Parity (P_o)	Odd Parity Checker (C_o)
0000	1	0
0001	0	0
0010	0	0
0011	1	0
0100	0	0
0101	1	0
0110	1	0
0111	0	0
1000	0	0
1001	1	0
1010	1	0
1011	0	0
1100	1	0
1101	0	0
1110	0	0
1111	1	0

Odd parity checker.

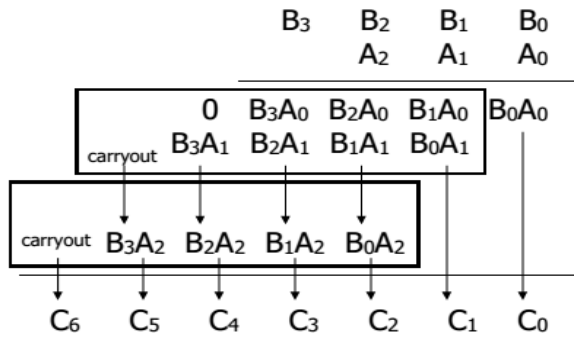
Binary Multiplier

$$\begin{array}{r} B_1 B_0 \\ A_1 A_0 \\ \hline A_0 B_1 \\ A_1 B_1 \\ \hline C_3 \end{array}$$

Circuit diagram



4-Bit By 3-Bit Binary Multiplier



We need 12 AND gates and two 4-bit adders to produce a product of 7 bits

Circuit diagram

