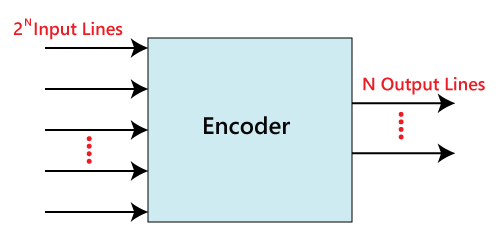
**1.ENCODERS AND MULTIPLEXERS**

1.1 Encoders:

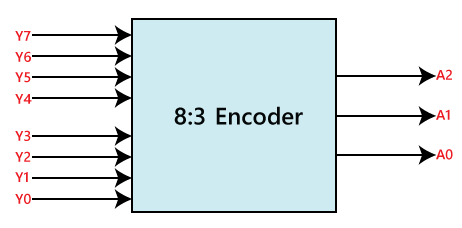
**Introduction:**

Encoders are specialized combinational logic circuits designed to transform binary data from 2^N input lines into a specific N-bit code represented on the output lines. This conversion process ensures that each unique combination of input lines activates only one specific output code, preventing duplication and ensuring the integrity of the data conversion. In a typical scenario, such as a 3-to-8 line encoder, with 3 input lines and 8 output lines, only one input line is active at any given time, generating a unique 3-bit binary code on the output lines. This methodology prevents plagiarism by creating distinct output codes for each input combination, ensuring the authenticity and uniqueness of the encoded data

**Fig.1.1 Logical symbol of Encoder**

**Octal to Binary Encoder**

The 8:3 line Encoder, also known as Octal to Binary Encoder, is a specific type of encoder that takes 8 inputs labeled Y0 to Y7 and produces three outputs represented by A0, A1, and A2. This encoder ensures that out of the 8 input lines, only one input line is activated (set to true) at a time. When activated, the encoder generates the corresponding binary code on the output side. The block diagram and truth table of this encoder illustrate its functionality, and it is represented using a specific logic symbol and truth table to depict its operation accurately.



**Fig. 1.2 Logical Symbol of 8:3 line Encoder**

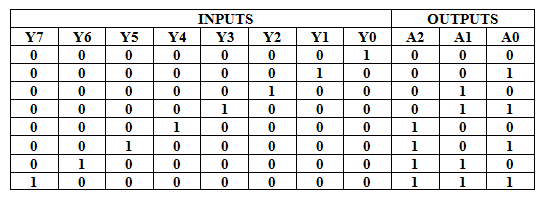
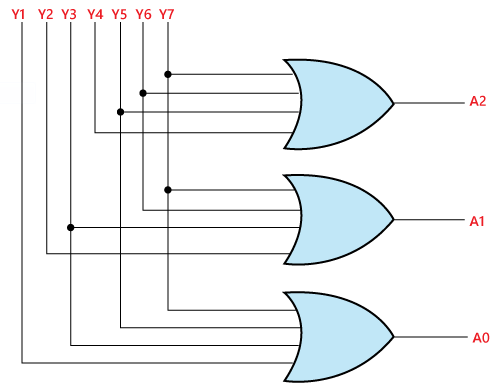


Table 1.1 Truth table for octal to binary conversion

Logical Expressions: A2=Y4+Y5+Y6+Y7 ; A1=Y2+Y3+Y6+Y7 A0=Y7+Y5+Y3+Y1

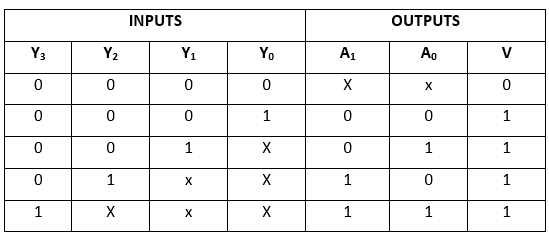
**Logical Circuit**



**Fig.1.3 Logical Circuit of Octal to Binary Conversion**

**Priority Encoder:**

In this type of encoder, each input (Y0 to Y3) is assigned a priority level. The encoder outputs, A0 and A1, indicate the active input with the highest priority. When multiple inputs are active simultaneously, the output corresponds to the binary code associated with the highest priority input. The truth table for this 4-to-2 line priority encoder determines the output based on the priority levels of the inputs.



By K map the following Logical expressions are obtained

A1=Y2+Y3 ; A0=Y3+Y2’Y1 ; V=Y0+Y1+Y2+Y3

**1.2 MULTIPLEXERS**

**1.2.1 Introduction:**

A multiplexer is a combinational circuit consists of 2n input lines and a single output line. In short, the multiplexer is a multi-input and single-output combinational circuit. The binary information is received from the input lines and will be sent to the output line. Since it is having select lines.On the basis of the values of the selection lines, one of these data inputs will be connected to the output. A multiplexer is also called as **Mux**.

There are different types of the multiplexer which are as follows:

## 2×1 Multiplexer:

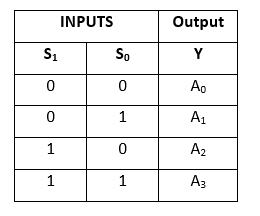
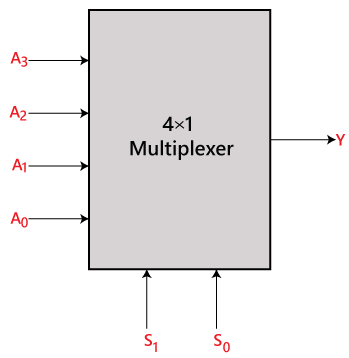
A 2×1 multiplexer is having 2 two inputs represented byA0, A1 and 1 selection line, i.e., S0 and single output represented by Y as shown. Based on the combination of inputs which are present at the selection line S0, one of the 2 inputs will be associated to the output. The block diagram and the truth table of the 2**×**1 multiplexer are given below.

### 

### Output equation is given byY=S0'.A0+S0.A1

## 4×1 Multiplexer:

In this type, there are total of four inputs which are represented by A0, A1, A2, and A3, 2 selection lines, i.e., S0 and S1 and single output, i.e Y. On the basis of the combination of inputs that are present at the selection lines S0 and S1, one of these 4 inputs are connected to the output. The block diagram and the truth table of the 4**×**1 multiplexer are given below.



**1.2.2 Using Multiplexers as a Boolean Function Generator**

A multiplexer is a digital circuit is used to select a single input from the multiple input lines. Since there are somedifficulties on multiplexer, which can be solved to get the desired circuit.

## Type #1

Problems on implementing the boolean expressions using a multiplexer. In this method, 3 variables are given(say P, Q, R), which are the selection inputs for the mux. For three selection inputs, the mux to be built was 2n = 23 = 8 : 1. So, in this method, the **type of mux can be decided by the given number of variables.**

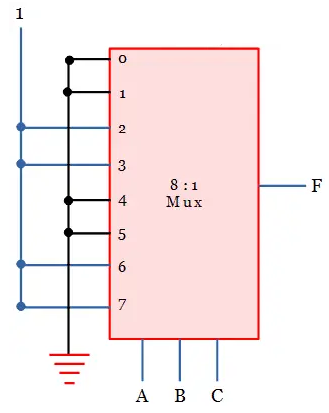
### Solved problem #1

**Implement the boolean expression F(A, B, C) = ∑ m(2, 3, 6, 7) using a multiplexer.**

**Solution:**

In the above problem there are 3 variables, hence 2n = 23 = 8 : 1 multiplexer. So, the mux has 8 input lines, 3 selection lines, and one output.

Therefore inputs associated with the minterms (2, 3, 6, 7) are connected to logic 1 and the remaining terms are connected to logic 0. The given input variables A,B,C are connected as three selection lines as shown.



## Type #2

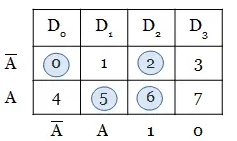
**Implement the boolean expression F(A, B, C) = ∑ m(0, 2, 5, 6) using 4 : 1 multiplexer.**

**Solution**:

In the givenboolean expression, there are 3 variables. We should use 23 : 1 = 8 : 1 multiplexer. But as per the question, it is to be implemented with 4 : 1 mux.

We know that for 4 : 1 multiplexer, there should be 2 selection lines. So, from the given 3 variables, we have to choose the 2 least significant variables (B, C) as selection line inputs.

Let us derive the four inputs of 4 : 1 multiplexer using the implementation table. The four inputs are listed in column-wise and all the minterms are written under the four inputs in 2 rows as shown below.



The minterms given in the boolean expression is circled and analyzed.

Thus the circuit can be drawn as below.

