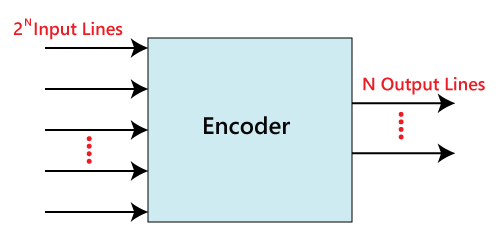
**1.ENCODERS AND MULTIPLEXERS**

1.1 Encoders:

**Introduction:**

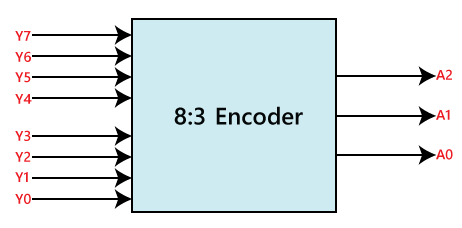
The encoders are combinational logic circuits changes the binary data into N output lines. The binary information is delivered in the form of 2N input lines. The output lines define the N-bit code for the binary information. Only one input line is activated at a time in encoders. Fig.1.1 illustrations general logical representation of Encoder.



**Fig.1.1 Logical symbol of Encoder**

**Octal to Binary Encoder**

Another name for 8:3 line Encoder is  **Octal to Binary Encoder**. This type of encoder is having 8 inputs Y0 toY7 and three outputs represented by A0, A1, and A2. In encoders out of 8-input lines always one input-line is set to true at a time to get the respective binary code in the output side. Figure shows the block diagram and the truth table of the 8 to 3 line encoder.The logic symbol and truth table are as shown below.



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

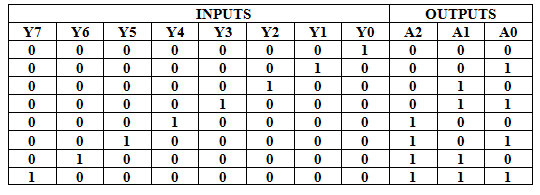
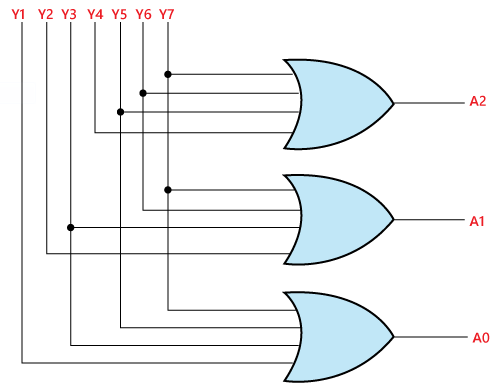


Table 1.1 Truth table for octal to binary conversion

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 a priority encoder each input has a priority level connected with it. The encoder outputs indicate the active input that has the highest priority. When an input with a high priority is declared, the other inputs with lower priority are ignored. The truth table for a 4-to-2 priority encoder is shown in Table 1.2. In this priority encoder, there are total of 4 inputs which are represented by Y0 to Y3 and two outputs represented by A0 and A1. The Y3 has highest and Y0 has lowest priority inputs. When more than one input is '1' at the same time, the output will be the (binary) code equivalent to the higher priority input. Below is the truth table of the 4 to 2 line priority encoder.

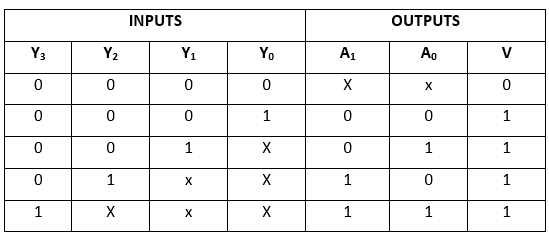


Table 1.2: Truth table of 4 bit priority encoder

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:

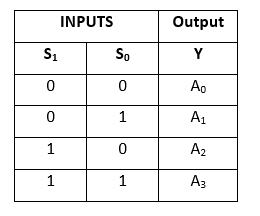
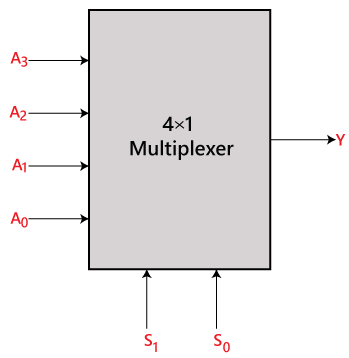
A 2×1 multiplexer is having 2 two inputs represented byA0, A1 and 1 selection line, i.e., S0 and single output represented by Y as shown. Based on the combination of inputs which are present at the selection line S0, one of the 2 inputs will be associated to the output. The block diagram and the truth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

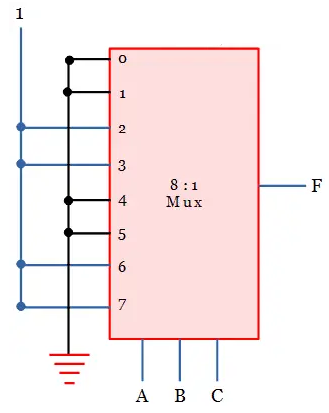
### Solved problem #1

**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

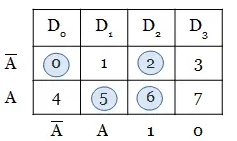
**Implement the boolean expression F(A, B, C) = ∑ m(0, 2, 5, 6) using 4 : 1 multiplexer.**

**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.

