CHAPTER

ENCODERS AND MULTIPLEXERS

Authors:

Sarita U and Vinoda S.

1.1 Encoders:

Introduction:

The combinational circuits that change the binary information into N output lines are known as **Encoders**. The binary information is passed in the form of 2^N input lines. The output lines define the N-bit code for the binary information. At a time, only one input line is activated for simplicity. The produced N-bit output code is equivalent to the binary information.Fig.1.1 shows general logical symbol of Encoder.

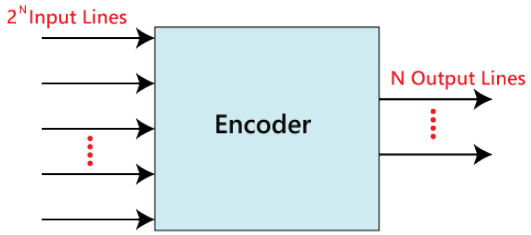


Fig.1.1 Logical symbol of Encoder

Octal to Binary Encoder

The 8 to 3 line Encoder is also known as **Octal to Binary Encoder**. In 8 to 3 line encoder, there is a total of eight inputs, i.e., Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, and Y_7 and three outputs, i.e., A_0, A1, and A_2. In 8-input lines, one input-line is set to true at a time to get the respective binary code in the output side. Below are the block diagram and the truth table of the 8 to 3 line encoder. Fig.1.2 shows the logical symbol of 8:3 encoder. Fig.1.3 shows the logical circuit.Table 1.1 shows truth table of octal to binary conversion.

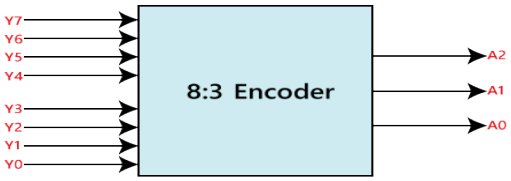


Fig. 1.2 Logical Symbol of 8:3 line Encoder

INPUTS								OUTPUTS		
Y7	**Y6**	**Y5**	**Y4**	**Y3**	**Y2**	**Y1**	**Y0**	**A2**	**A1**	**A0**
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**

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

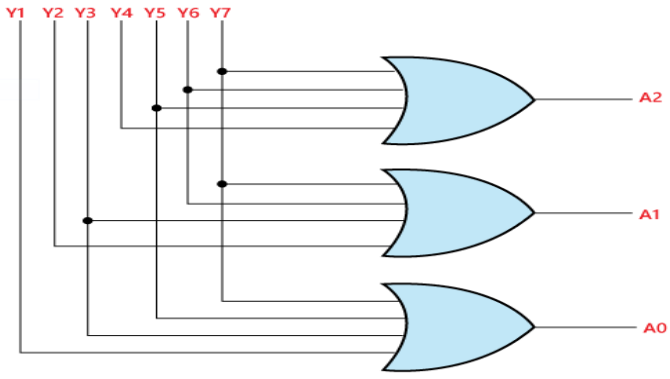


Fig.1.3 Logical Circuit of Octal to Binary Conversion

Priority Encoder:

Another useful class of encoders is based on the priority of input signals. In a priority encoder each input has a priority level associated with it. The encoder outputs indicate the active input that has the highest priority. When an input with a high priority is asserted, 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, i.e., Y_0, Y_1, Y_2, and Y_3, and two outputs, i.e., A_0 and A_1. The Y_3 has high and Y_0 has low priority inputs. When more than one input is '1' at the same time, the output will be the (binary) code corresponding to the higher priority input. Below is the truth table of the 4 to 2 line priority encoder.

INPUTS				OUTPUTS		
Y_3	Y_2	Y_1	Y_0	A_1	A_0	V
0	0	0	0	X	x	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	x	X	1	0	1
1	X	x	X	1	1	1

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:

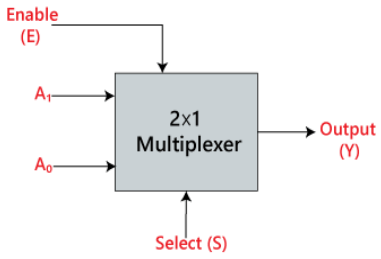
A multiplexer is a combinational circuit that has 2^n input lines and a single output line. Simply, the multiplexer is a multi-input and single-output combinational circuit. The binary information is received from the input lines and directed to the output line. On the basis of the values of the selection lines, one of these data inputs will be connected to the output. A multiplexer is also treated as **Mux**.

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

1. 2×1 Multiplexer:

In 2×1 multiplexer, there are only two inputs, i.e., A_0 and A_1, 1 selection line, i.e., S_0 and single outputs, i.e., Y. On the basis of the combination of inputs which are present at the selection line S_0, one of these 2 inputs will be connected to the output. The block diagram and the truth table of the 2×1 multiplexer are given below.

Block Diagram:



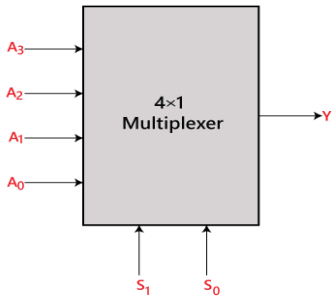
Truth Table:

INPUTS	Output
S_0	Y
0	A_0
1	A_1

The logical expression of the term Y is as follows: $Y=S_0'.A_0+S_0.A_1$

2. 4×1 Multiplexer:

In the 4×1 multiplexer, there is a total of four inputs, i.e., A_0, A_1, A_2, and A_3, 2 selection lines, i.e., S_0 and S_1 and single output, i.e., Y. On the basis of the combination of inputs that are present at the selection lines S_0 and S_1, 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.



INPUTS		Output
S_1	S_0	Y
0	0	A_0
0	1	A_1
1	0	A_2
1	1	A_3

1.2.2 Using Multiplexers as a Boolean Function Generator

A multiplexer is a digital circuit, which is used to select a single input from the multiple input lines. There are several problems on multiplexer, which are solved to get the desired circuit.

Type #1

Let us solve some 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 $2^n = 2^3 = 8 : 1$. So, in this method, the **type of mux can be decided by the given number of variables.**

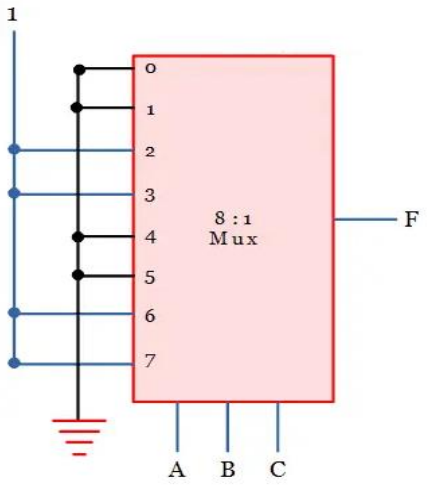
Solved problem #1

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

Solution:

There are 3 variables in the given expression, hence $2^n = 2^3 = 8 : 1$ multiplexer. So, the mux has 8 input lines, 3 selection lines, and one output.

The inputs, corresponding to the minterms (2, 3, 6, 7) are connected to logic 1 and the remaining terms to logic 0(grounded). The given input variables are connected as three selection lines.



Type #2

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

Solution:

In the given boolean expression, there are 3 variables. We should use 2^3 : 1 = 8 : 1 multiplexer. But as per the question, it is to be implemented with 4 : 1 mux.

For 4 : 1 multiplexer, there should be 2 selection lines. So from the given 3 variables, the 2 least significant variables(B, C) are used 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.

	D_0	D_1	D_2	D_3
$\overline{A}$	0	1	2	3
A	4	5	6	7
	$\overline{A}$	A	1	0

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

Thus the circuit can be drawn as below.