

Multiplexers (Data Selectors) :-

- The term 'multiplex' means 'many into one'. Multiplexing is the process of transmitting a large number of information over a single line/channel.
- A digital multiplexer (MUX) is a combinational circuit that selects one digital information from several sources and transmits the selected information on a single output line.
- Multiplexer is also called a data selector since it selects one of many inputs and passes the information to the output.
- The multiplexer has several data-input lines. The selection of a particular input line is controlled by a set of selection lines.
- The block diagram of a multiplexer with  $n$  input lines,  $m$  select lines and one output line is shown in fig 1(a).

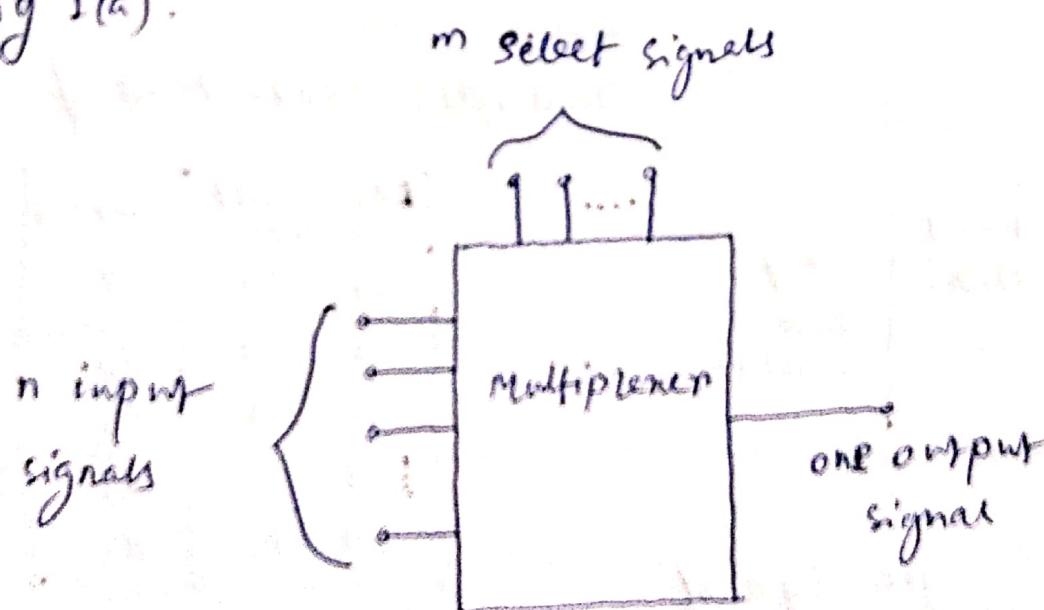


Fig 1(a) : Block diagram of a multiplexer .

→ The selection lines decide the number of input lines of a particular multiplexer.

If the number of  $n$  input lines is equal to  $2^m$ , then  $m$  select lines are required to select one of the  $n$  input lines. For example, to select 1 out of 4 input lines, two select lines are required; to select 1 of 8 input lines, three select lines are required and so on.

#### □ Basic 4-input Multiplexer:

The logic symbol of a 4-to-1 multiplexer is shown in fig 1(b). It has four data input lines ( $D_0 - D_3$ ), a single output line ( $Y$ ) and two select lines ( $S_1$  and  $S_0$ ) — to select one of the four input lines. The truth table for a 4-to-1 multiplexer is shown in table 1(a).

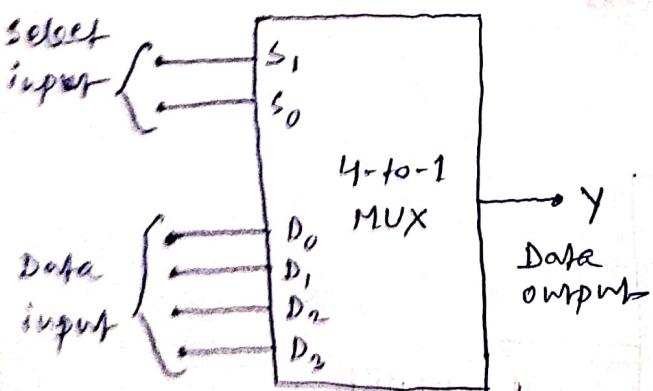


Fig 1(b): Logic symbol of 4-to-1 MUX

Table 1(a): Truth table of 4-to-1 MUX

Date Select Inputs	Output	
$S_1$	$S_0$	$Y$
0	0	$D_0$
0	1	$D_1$
1	0	$D_2$
1	1	$D_3$

From the truth table of table 1(a), logical expression for the output in terms of the data input and the select inputs can be derived as follows:

If  $s_1 = 0$  and  $s_0 = 0$ , the output  $y = D_0$

therefore,  $y = D_0 \bar{s}_1 \bar{s}_0 = D_0$  when  $s_1 s_0 = 00$

If  $s_1 = 0$  and  $s_0 = 1$ , the output  $y = D_1$ ,

~~Therefore~~ or  $y = D_1 \bar{s}_1 s_0 = D_1$  when  $s_1 s_0 = 01$

If  $s_1 = 1$  and  $s_0 = 0$ , the output  $y = D_2$

or  $y = D_2 s_1 \bar{s}_0 = D_2$  when  $s_1 s_0 = 10$

If  $s_1 = 1$  and  $s_0 = 1$ , the output  $y = D_3$

or  $y = D_3 s_1 s_0 = D_3$  when  $s_1 s_0 = 11$

If the above terms are ORed, then the final expression for the data output is given by

$$y = D_0 \bar{s}_1 \bar{s}_0 + D_1 \bar{s}_1 s_0 + D_2 s_1 \bar{s}_0 + D_3 s_1 s_0$$

→ Using the above expression, the 4-to-1 multiplexer can be implemented using two NOT gates, four 3-input AND gates and one 4-input OR gate. as shown in fig 1(c).

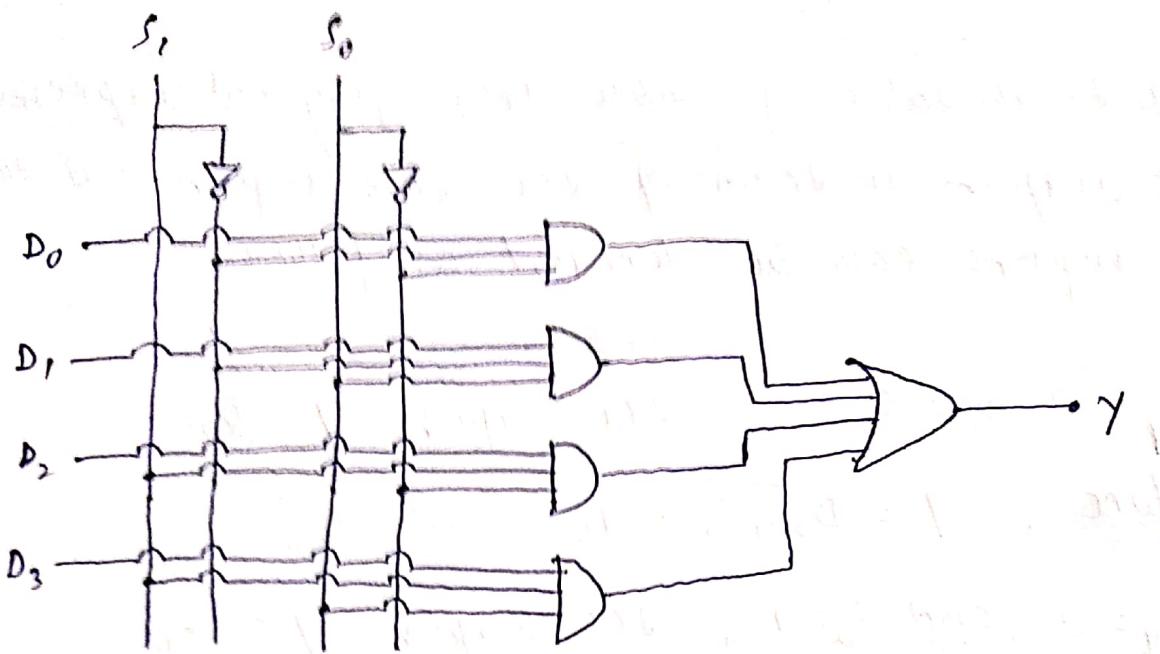


Fig 1(c) : A logic diagram of a 4-to-1 MUX.

#### □ 8-to-1 Multiplexer :-

- There are eight data inputs ( $D_0 - D_7$ ), three select input lines ( $S_2 - S_0$ ) and a single output ( $Y$ ). It also has an enable input  $\bar{E}$  and provides both normal and inverted outputs (i.e.  $Y$  and  $\bar{Y}$ ). Since  $2^3 = 8$ , three bits are required to select any one of the eight data bits.
- When  $\bar{E} = 0$ , the select inputs  $S_2, S_1, S_0$  will select one of the data input to pass through the output  $Y$ . When  $\bar{E} = 1$ , the multiplexer is disabled. The logic symbol of IC 74151 is shown in fig 1(d) and its operation of this IC is summarised in truth table 1(b). logic diagram is shown in fig 1(c).

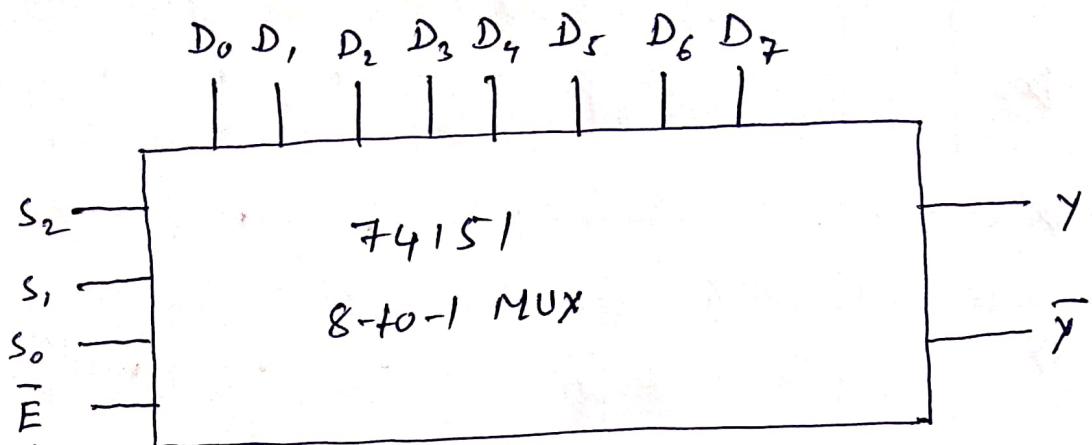


Fig 1(d) : Logic symbol of IC 74151 8-to-1 MUX.

Table 1(b) : Truth table of IC 74151 - 8-to-1 Multiplexer.

Inputs				outputs	
$\bar{E}$	$S_2$	$S_1$	$S_0$	$Y$	$\bar{Y}$
1	X	X	X	1	$\bar{D}_0$
0	0	0	0	$D_0$	$\bar{D}_0$
0	0	0	1	$D_1$	$\bar{D}_1$
0	0	1	0	$D_2$	$\bar{D}_2$
0	0	1	1	$D_3$	$\bar{D}_3$
0	1	0	0	$D_4$	$\bar{D}_4$
0	1	0	1	$D_5$	$\bar{D}_5$
0	1	1	0	$D_6$	$\bar{D}_6$
0	1	1	1	$D_7$	$\bar{D}_7$

## Implementation of higher order Multiplexer:

- It is possible to implement higher order multiplexers, i.e. multiplexers with more number of inputs, using lower order multiplexers, i.e. multiplexers with lesser number of inputs.

For example, a 16-to-1 multiplexer can be implemented by either using two 8-to-1 multiplexers or four 4-to-1 multiplexers.

Example: Implement a 16-to-1 multiplexer using two 8-to-1 multiplexers.

- ⇒ A 16-to-1 multiplexer can be implemented using two IC 74151 8-to-1 multiplexers and one 2-input OR gate as shown in fig. 1(e).
- Here, to select one of the 16 inputs, four select lines ( $S_3, S_2, S_1, S_0$ ) are required. Among the four select lines, the least significant three select lines ( $S_2, S_1, S_0$ ) are connected with three select inputs of both the multiplexer ICs. The most significant select line  $S_3$  is connected directly to the  $\bar{S}$  input of MUX 1 while the same is connected through an inverted to the  $\bar{S}$  input of MUX 2.

Therefore, when  $S_3=0$ , MUX1 is selected and the inputs ( $D_0$  to  $D_7$ ) are multiplexed to the output Z and MUX2 is disabled.

When  $S_3=1$ , the MUX1 is disabled while MUX2 is enabled and the inputs ( $D_8$  to  $D_{15}$ ) are multiplexed to the output Z. Also, note that the outputs of MUX1 and MUX2 (i.e.  $y_1$  and  $y_2$ ) are ORed using an OR gate to generate output Z.

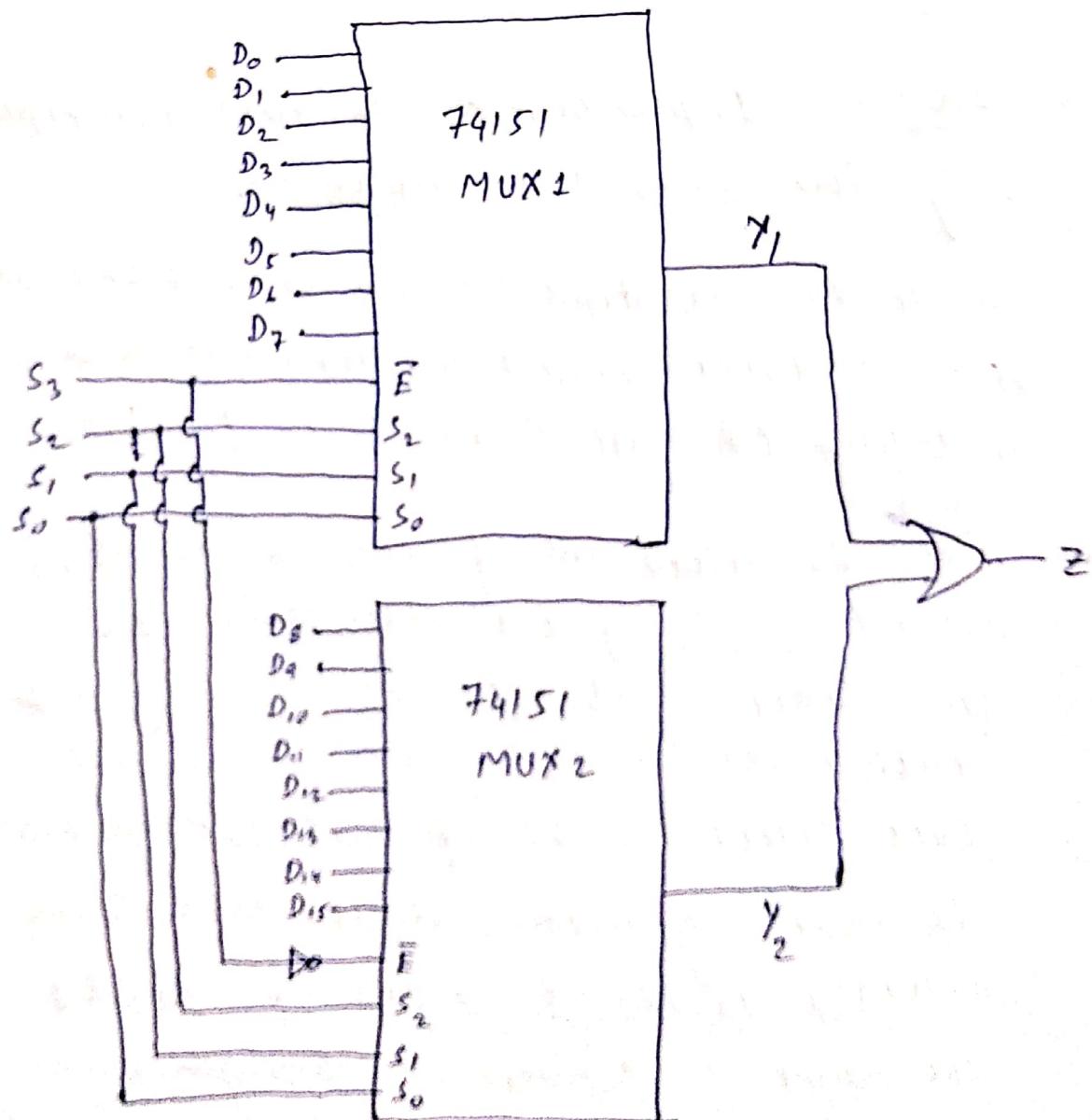


Fig 1(e): A 16-to-1 multiplexer using two IC 74151 -  
8-to-1 Multiplexers.

## Implementation Of Boolean expression using Multiplexers:

- Any boolean expression can be easily implemented using a multiplexer. If a boolean expression has  $(n+1)$  variables, then  $n$  of these variables can be connected to the select lines of the multiplexer. The remaining single variable ~~as~~ along with the constants 1 and 0 is used as the input of the multiplexer.

For example, if  $A$  is a single variable, then the inputs of the multiplexer are ~~are~~  $A, \bar{A}, 1$  and 0.

- In general, a boolean expression of  $(n+1)$  variables can be implemented using a multiplexer with  $2^n$  inputs.

To demonstrate this process, consider the function

$$F(A, B, C, D) = \sum(0, 1, 3, 4, 8, 9, 15)$$

As the given function is a four variable function, we need a multiplexer with three select lines and eight inputs ( $2^3$ ).

- Apply variables B, C and D to the select lines. The procedure for implementing the function are:

- List the inputs of the multiplexer and
- List ~~under~~ under them all the minterms are associated with  $\bar{A}$  and the second half with  $A$ . in two rows as shown in table 1.

The first half of the minterms are associated with  $\bar{A}$  and the second half with  $A$ . The given function is implemented by circling the minterms of the function and applying the following rules to find the values for the inputs of the multiplexer.

1. If both the minterms in a column are not circled, apply 0 to the corresponding input.
2. If both the minterms in a column are circled, apply 1 to the corresponding input.
3. If the bottom minterm is circled and the top is not circled, Apply  $A$  to the input.
4. If the top minterm is circled and the bottom is not circled, Apply  $\bar{A}$  to the input.

Table 1: procedure for implementation of the function

	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
$\bar{A}$	(0)	(1)	2	(3)	(4)	5	6	7
$A$	(8)	(9)	10	11	12	13	14	(15)
	1	1	0	$\bar{A}$	$\bar{A}$	0	0	$A$

- Now using the procedure and the table, the given function can be implemented using an 8-to-1 multiplexer as shown in fig 1.
- It is not necessary to choose the most significant variable as an input to the multiplexer. one can choose any one of the variables as an input and

accordingly the multiplexer implementation table has to be modified.

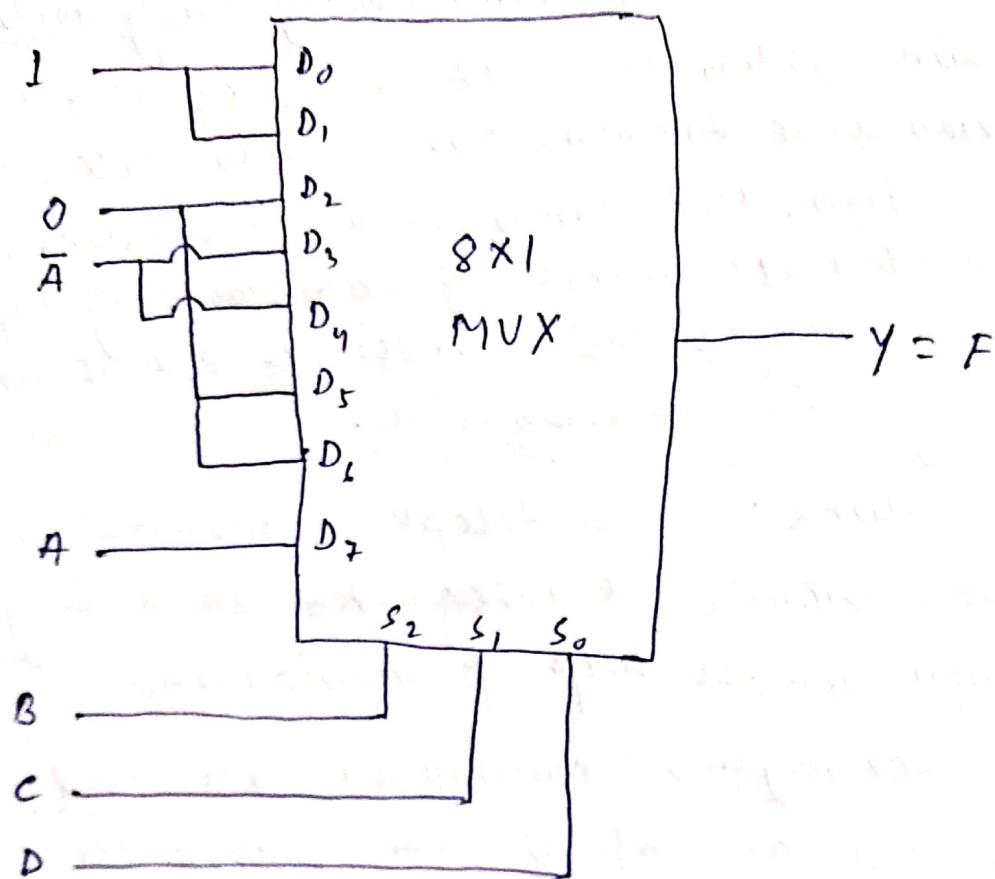


Fig 1: Implementation of  $F(A, B, C, D) = \sum(0, 1, 3, 4, 5, 9, 10)$  using 8-to-1 MUX.

④ Assignment Question:

Implement the following function using a multiplexer.

$$F(A, B, C) = \sum(1, 3, 5, 6).$$

## Applications of multi multiplexers: ~

1. Communication system: ~ The efficiency of the communication system can be increased considerably using multiplexers. communication system is a set of systems that enable communication like transmission system, relay and tributary station, and communication network. 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~~ audio signals are integrated on a single for transmission with the help of multiplexers.
3. computer memory: ~ Multiplexers are used to implement huge amount of memory into the computer, at the same time reduces the 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 spacecraft to the ground system using the GPS (global positioning System) satellites.
5. Parallel-to-Serial converter: ~ In order to transmit the information over long distances, the parallel arrangement is undesirable as it requires a large number of transmission line. Therefore, data in parallel form is converted to serial form using multiplexers.

## Demultiplexers (Data Selector Distributors) :-

- The word 'demultiplexing' means 'one into many'. Demultiplexing is the process of taking information from one input and transmitting the same over one of several outputs.
- A demultiplexer is a logic circuit that receives information on a single input and transmits the same information over one of several ( $2^n$ ) output lines.
- The block diagram of a demultiplexer which is opposite to a multiplexer in its operation is shown in fig 2(a). The circuit has one input signal,  $n$  select signals and  $n$  output signals. The select inputs determine to which output the data input will be connected. As the serial data is changed to parallel data, i.e. the inputs caused to appear on one of the  $n$  output lines, the demultiplexer is also called a distributor or a serial-to-parallel converter.

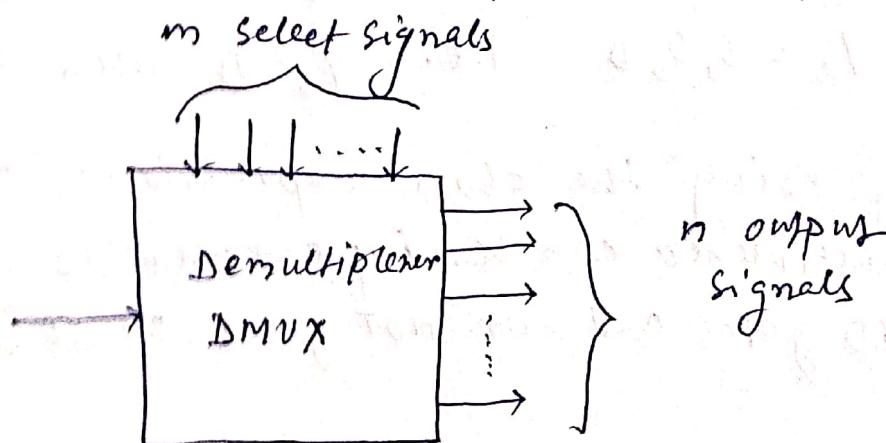


Fig 2(a): Block diagram of Demultiplexer.

- Q) 1-to-4 Demultiplexer: ~
- A 1-to-4 demultiplexer has a ~~big~~ single input (D), four outputs ( $y_0$  to  $y_3$ ) and two select lines ( $s_1$  and  $s_0$ ). The truth table of the 1-to-4 demultiplexer is shown in table 1(a).

Table 1(a): Truth table of 1-to-4 demultiplexer.

Data inputs	Select inputs		outputs			
D	$s_1$	$s_0$	$y_3$	$y_2$	$y_1$	$y_0$
D	0	0	0	0	0	D
D	0	1	0	0	D	0
D	1	0	0	D	0	0
D	1	1	D	0	0	0

From the truth table, the expressions for outputs can be written as follows:

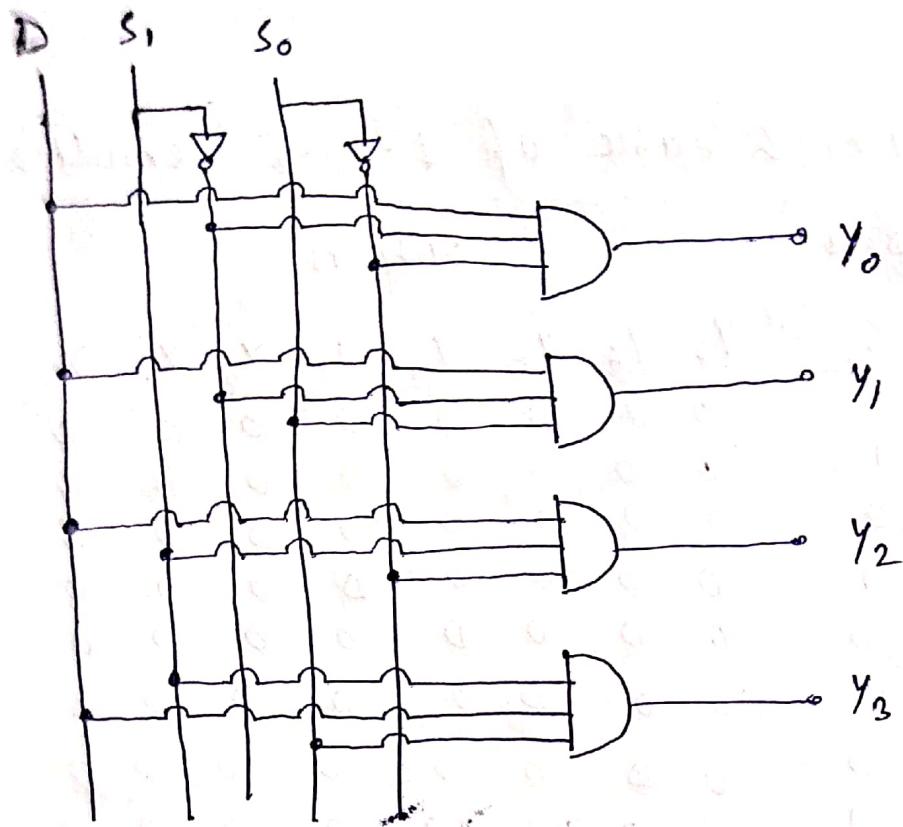
$$y_0 = \bar{s}_1 \bar{s}_0 D \quad \text{i.e. } y_0 = D \text{ when } s_1 = 0 \text{ and } s_0 = 0$$

$$y_1 = \bar{s}_1 s_0 D \quad \text{i.e. } y_1 = D \text{ when } s_1 = 0 \text{ and } s_0 = 1$$

$$y_2 = s_1 \bar{s}_0 D \quad \text{i.e. } y_2 = D \text{ when } s_1 = 1 \text{ and } s_0 = 0$$

$$y_3 = s_1 s_0 D \quad \text{i.e. } y_3 = D \text{ when } s_1 = 1 \text{ and } s_0 = 1$$

Now, using the above expressions, a 1-to-4 demultiplexer can be implemented using four 3-input AND gates and two NOT gates as shown in fig 2(b)



Q 2(b): Logic diagram of 1-to-4 demultiplexer.

### 1-to-8 demultiplexer:

- It has a single input (D), eight outputs ( $y_0$  to  $y_7$ ) and three select inputs ( $s_2$ ,  $s_1$ , and  $s_0$ ). It distributes one input line to eight output lines based on the select inputs. The truth table of 1-to-8 demultiplexer is shown in table 2(6).
- Now, from the truth table, the expressions for eight outputs can be written as follows:

$$y_0 = \bar{s}_2 \bar{s}_1 \bar{s}_0 D$$

$$y_1 = \bar{s}_2 \bar{s}_1 s_0 D$$

$$y_2 = \bar{s}_2 s_1 \bar{s}_0 D$$

$$y_3 = \bar{s}_2 s_1 s_0 D$$

$$y_4 = s_2 \bar{s}_1 \bar{s}_0 D$$

$$y_5 = s_2 \bar{s}_1 s_0 D$$

$$y_6 = s_2 s_1 \bar{s}_0 D$$

$$y_7 = s_2 s_1 s_0 D$$

Table 2(b) : Truth table of 1-to-8 demultiplexers.

Data input	select inputs $S_2 S_1 S_0$	outputs $Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0$							
D	0 0 0	0	0	0	0	0	0	0	D
D	0 0 1	0	0	0	0	0	0	D	0
D	0 1 0	0	0	0	0	0	D	0	0
D	0 1 1	0	0	0	0	D	0	0	0
D	1 0 0	0	0	0	D	0	0	0	0
D	1 0 1	0	0	D	0	0	0	0	0
D	1 1 0	0	D	0	0	0	0	0	0
D	1 1 1	D	0	0	0	0	0	0	0

## Decoders :-

- A decoder is similar to demultiplexer but without any ~~select~~ input. A decoder is a logic circuit that converts an n-bit binary input code (data) into  $2^n$  output lines, such that each output lines will be activated for only one of the possible combination of inputs.
- In a decoder, the number of output is greater than the number of inputs. Here, it is important to note that if the number of inputs and outputs are equal in a digital system then it can be called converters, e.g. BCD to Excess-3 code, binary to Gray and Gray to binary converters.

## 3-to-8 decoder:-

- A 3-to-8 decoder has three inputs ( $A, B, C$ ) and eight outputs ( $D_0$  to  $D_7$ ). Based on the 3 inputs, one of the eight outputs is selected.  
The truth table for 3-to-8 decoder is shown in Table 3(a).
- From the above truth table, it is clear that only one of eight ( $D_0$  to  $D_7$ ) outputs is selected based on the three select inputs. The logic

Table 3(a) : Truth table of 3-to-8 decoder.

Inputs			Outputs							
A	B	C	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
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

Also from the truth table, it is clear that only the logic expressions for the outputs can be written as follows:

$$D_0 = \bar{A}\bar{B}\bar{C} \quad D_1 = \bar{A}\bar{B}C \quad D_2 = \bar{A}BC \quad D_3 = \bar{A}B\bar{C}$$

$$D_4 = A\bar{B}\bar{C} \quad D_5 = A\bar{B}C \quad D_6 = AB\bar{C} \quad D_7 = ABC$$

→ This decoder can be used for decoding any 3-bit code to provide eight outputs, corresponding to eight different combinations of the input code.

→ Enable inputs: Some decoders have one or more enable inputs which are used to control the operation of the decoder. With the enable line held HIGH, the decoder functions normally and the input code, A, B and C will determine which output is HIGH.

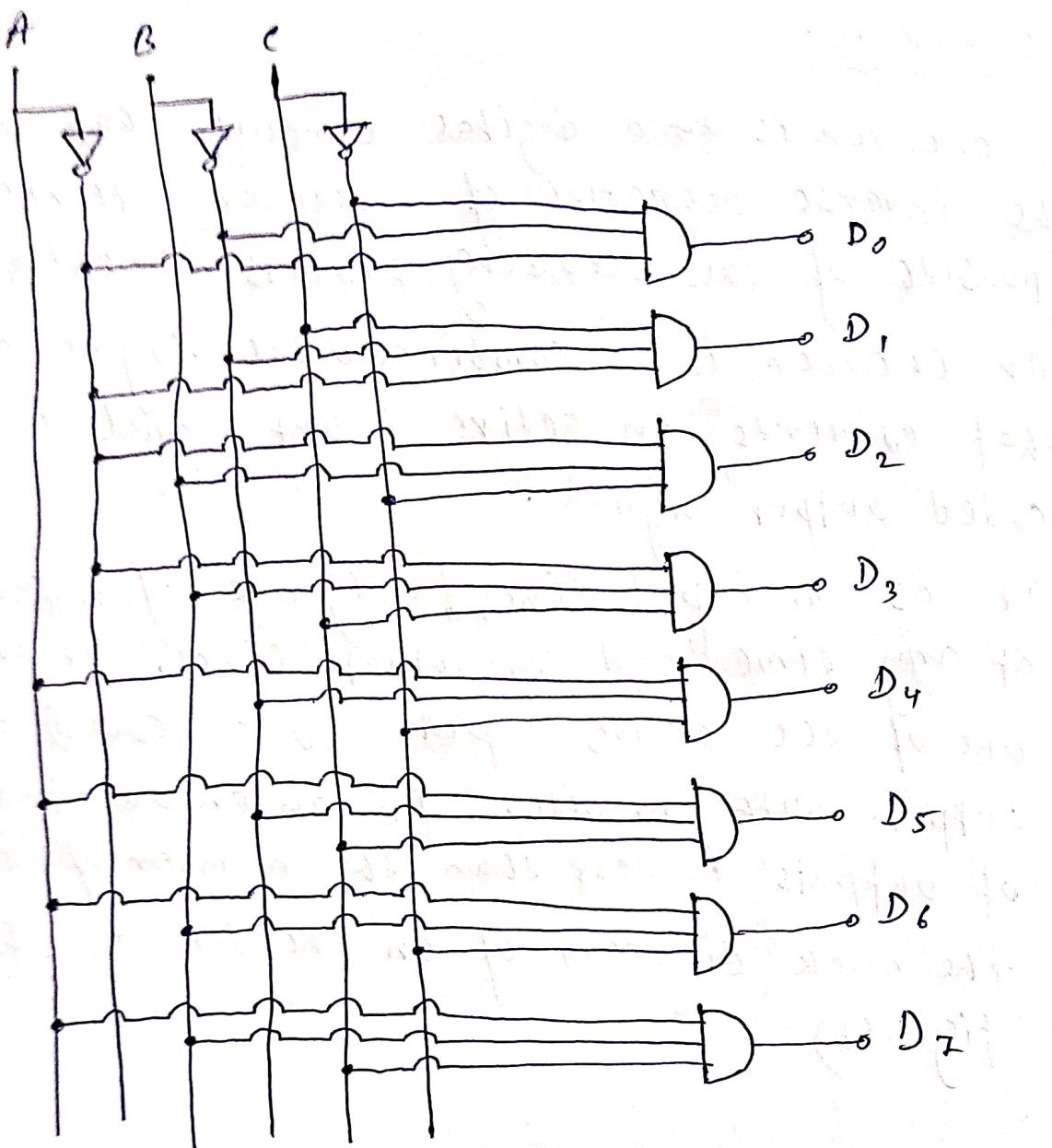


Fig 3(a) : logic diagram of 3-to-8 decoder.

#### Applications of decoders:

- (1) Decoders are used in counter systems
- (2) They are used in analog to digital converters
- (3) Decoder outputs can be used to drive a display system.

## □ Encoders:

- An encoder is a digital circuit that performs the inverse operation of a decoder. Hence the opposite of the decoding process is called encoding.
- An encoder is a combinational logic circuit that converts an active input signal into a coded output signal.

It has  $n$  input lines, only one of which is active at any time and  $m$  output lines. It encodes one of the active inputs to a coded binary output with  $m$  bits. In an encoder, the number of outputs is less than the number of inputs.

The block diagram of an encoder is shown in Fig 3(b).

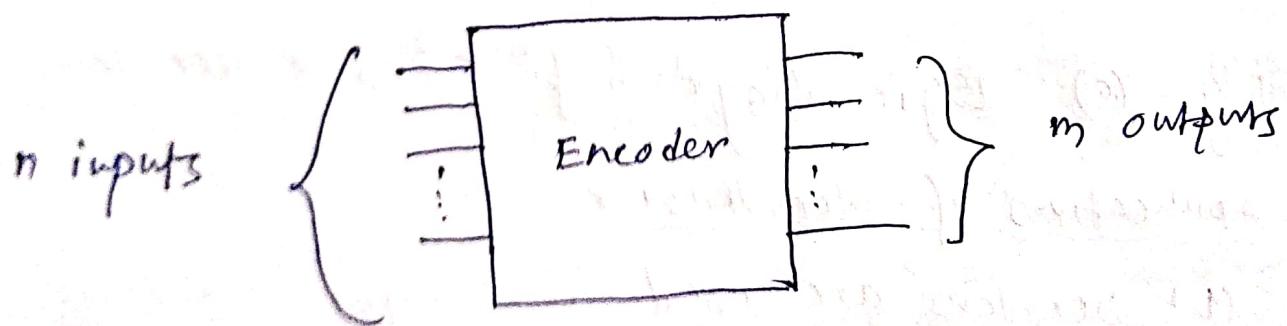


Fig 3(b) : Block diagram of an encoder.

## Example of Encoder:-

### ② Decimal-to-BCD encoder :-

A decimal-to-BCD encoder is one with ten inputs corresponding to ten decimal digits (0 to 9) and four outputs ( $A, B, C, D$ ) representing the BCD value of input decimal digit.

Table 3(b): Truth table of decimal-to-BCD encoder

Decimal inputs										BCD outputs			
0	1	2	3	4	5	6	7	8	9	A	B	C	D
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

## Parity generators / checkers :

- When digital data is transmitted from one ~~to~~ location to another, it is necessary to know at the receiving end whether the received data is free of error.
- A simple form of error detection is achieved by adding an extra bit to the transmitted word.
- There are two types of parity bits, namely even parity and odd parity.

In an even parity system, the parity bit added to the word to be transmitted is chosen so that the number of 1s in the modified word is even. Odd parity means an  $n$ -bit input has an odd number of 1s.

## Design of parity checkers :

Exclusive OR gate (XOR) is ideal for checking the parity of a binary number because they produce an output when the input has an odd number of 1s. Therefore, an even-parity input to an XOR gate produces a low output, while an odd-parity input produces a high output.

To check the parity of given  $n$ -bit number,  $(n-1)$  two input XOR gates can be cascaded.

