Digital Logic Design (EL-227)

LABORATORY MANUAL

Spring-2021

LAB 09

Multiplexer And demultiplexer

STUDENT NAME ROLL NO SEC

INSTRUCTOR SIGNATURE& DATE

MARKS AWARDED: /03

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), KARACHI

Date: 3-May-2021

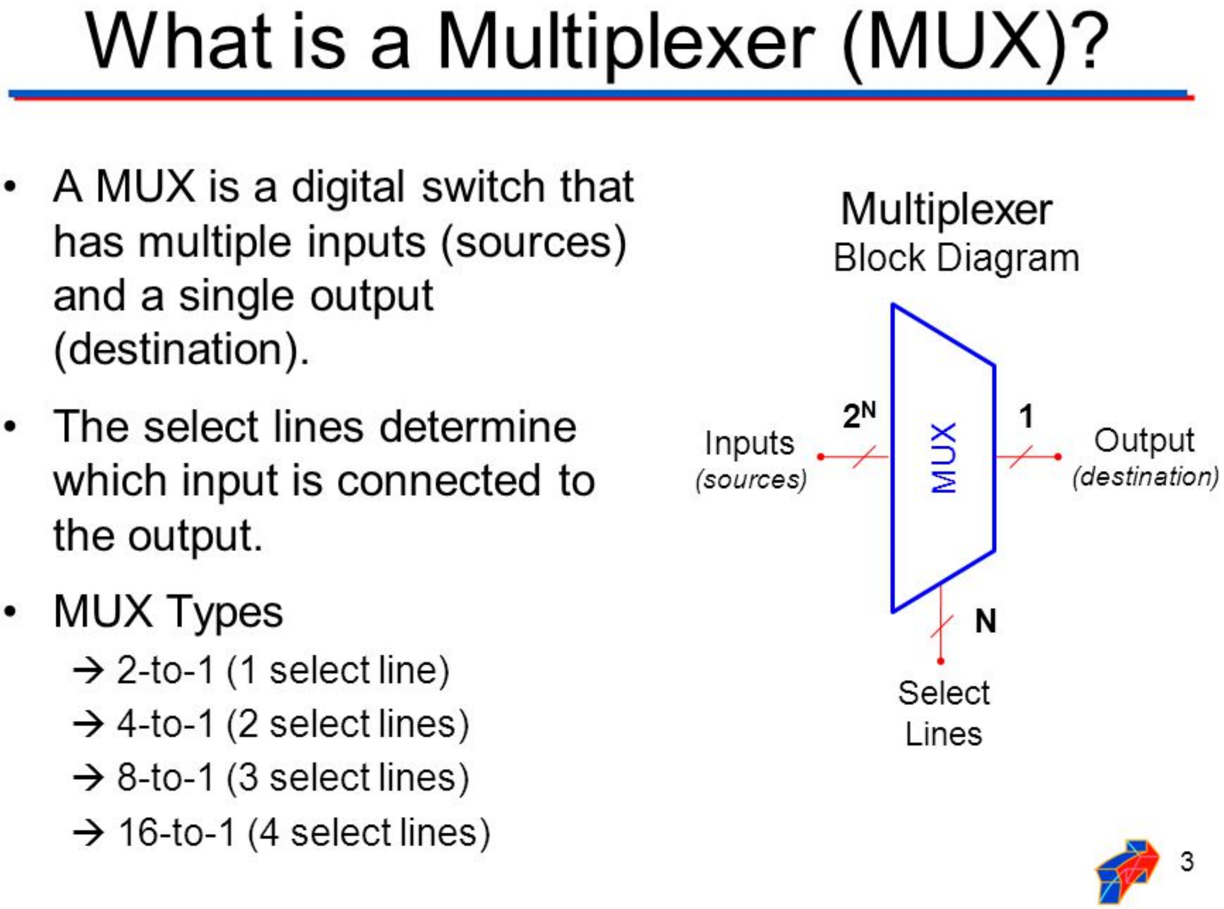
Lab Session 09:Multiplexer

OBJECTIVES:

- To study the concept of multiplexers
- To learn the implementation of Boolean function using multiplexer To learn
- how to implement large multiplexer using small size multiplexers
- Parallel Multiplexing using 74LS151

APPARATUS: Logic trainer, Logic probe

COMPONENTS: ICs 74LS08, 74LS32, 74LS04, 74LS86, 74LS02



What is a Multiplexer (MUX)?

- A MUX is a digital switch that has multiple inputs (sources) and a single output (destination).
- The select lines determine which input is connected to the output.
- MUX Types
  - → 2-to-1 (1 select line)
  - → 4-to-1 (2 select lines)
  - → 8-to-1 (3 select lines)
  - → 16-to-1 (4 select lines)

Multiplexer is a combinational circuit that has maximum of 2^n data inputs, 'n' selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines.

Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination will select only one data input. Multiplexer is also called as **Mux**.

Multiplexer: many into one

A multiplexer is a device that allows data information from several sources to be routed onto a single line for transmission to a common destination. It has several input lines and a single output line. It also has data-select inputs, which permit digital data on any one of the inputs to be switched to the output line. A multiplexer is also called a data selector because of this ability to select which data input is connected o the output. Normally there are 2n input lines and n selection lines whose bit combination determine which input is selected.

A 74LS151 has eight inputs that can be individually selected by three select lines. The output is connected to the input line selected by the binary value on the three select lines. If the three select lines are all zeros, then input line "0" is selected and connected to the output line.

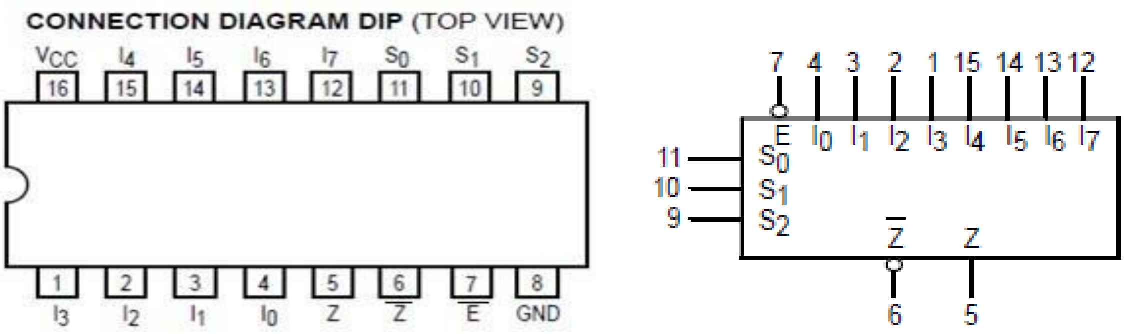


Fig 1: Pin's Description

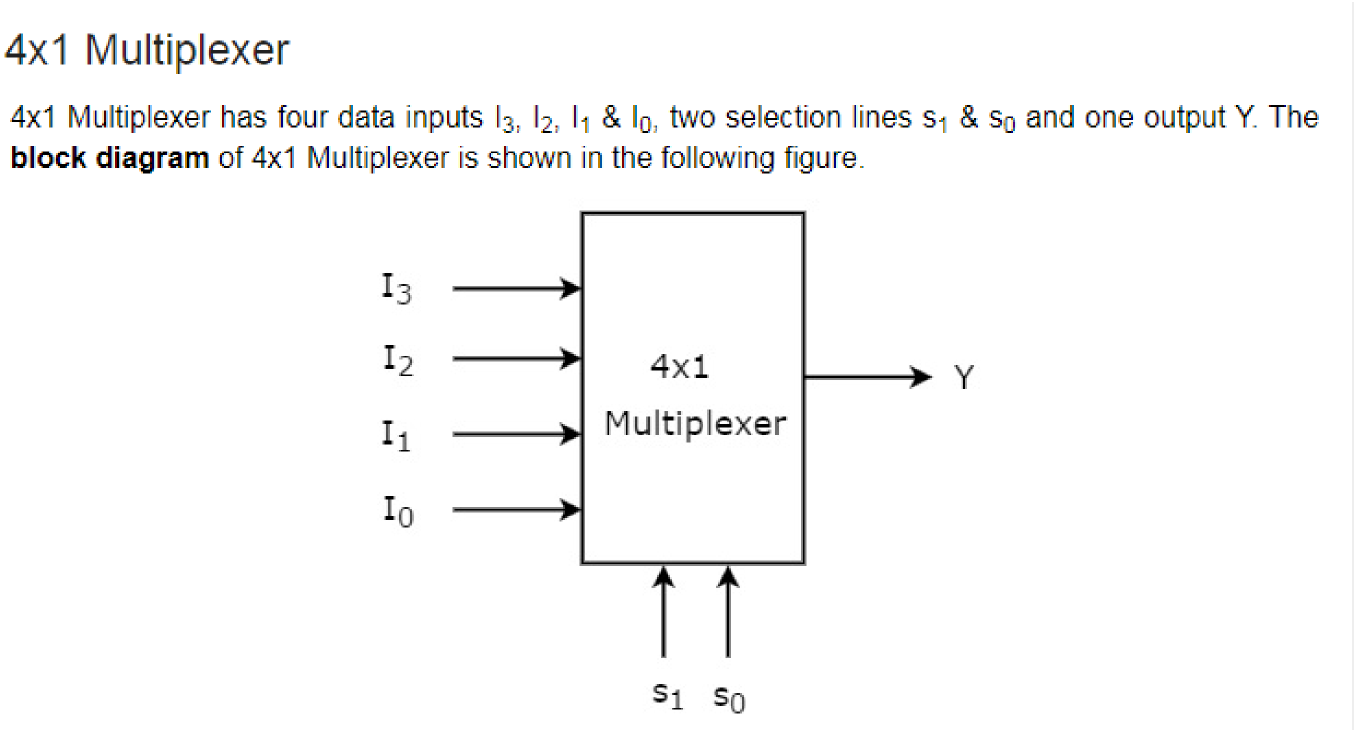
Applications of Multiplexer:

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.

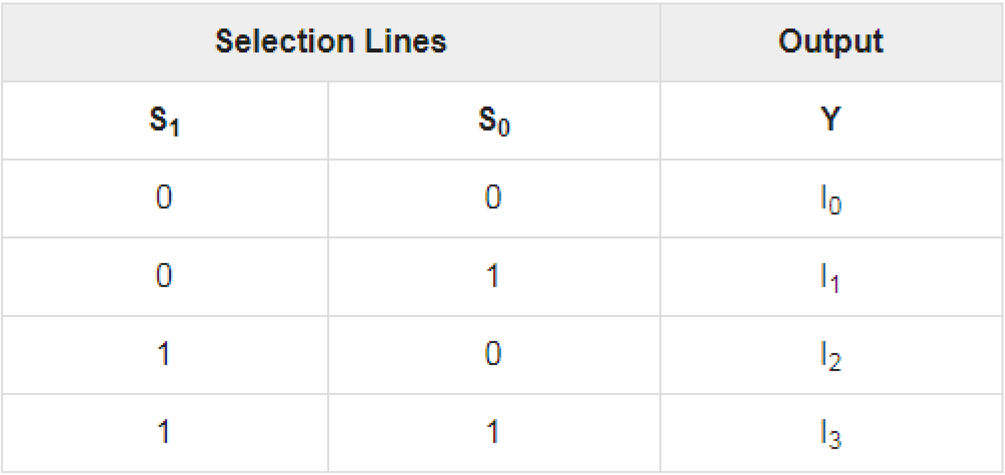
3. **Computer Memory** - Multiplexers are used to implement huge amount of memory into the computer, at the same time reduces the number of copper lines required to connect the memory to other parts of the computer circuit.



4x1 Multiplexer

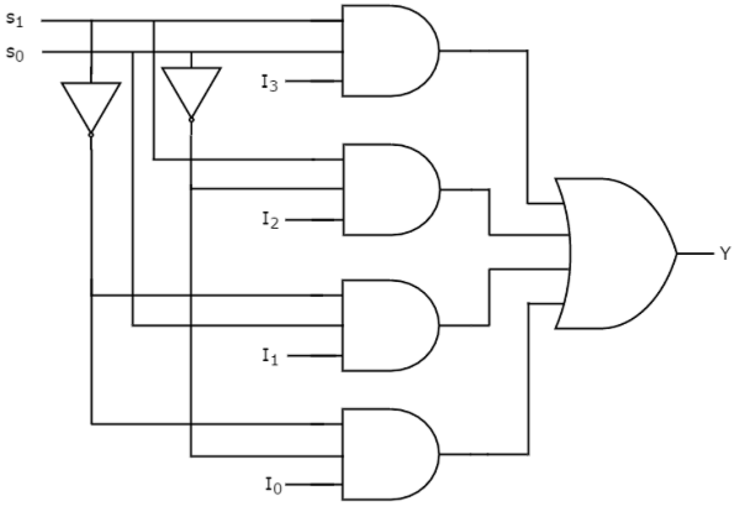
4x1 Multiplexer has four data inputs I_3, I_2, I_1 & I_0, two selection lines s_1 & s_0 and one output Y. The **block diagram** of 4x1 Multiplexer is shown in the following figure.

One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines. **Truth table** of 4x1 Multiplexer is shown below.



Selection Lines		Output
S_1	**S_0**	**Y**
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

We can implement this Boolean function using Inverters, AND gates & OR gate. The **circuit diagram** of 4x1 multiplexer is shown in the following figure.



Implementation of Higher-order Multiplexers.

Now, let us implement the following two higher-order Multiplexers using lower-order Multiplexers.

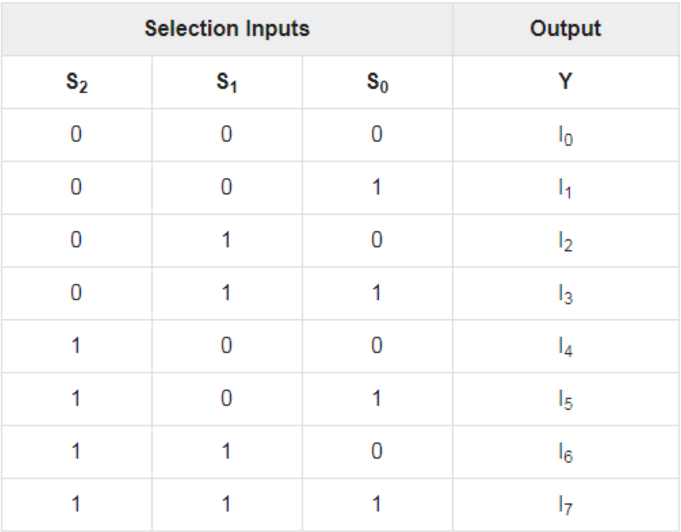
- 8x1 Multiplexer
- 16x1 Multiplexer

8x1 Multiplexer

In this section, let us implement 8x1 Multiplexer using 4x1 Multiplexers and 2x1 Multiplexer. We know that 4x1 Multiplexer has 4 data inputs, 2 selection lines and one output. Whereas, 8x1 Multiplexer has 8 data inputs, 3 selection lines and one output.

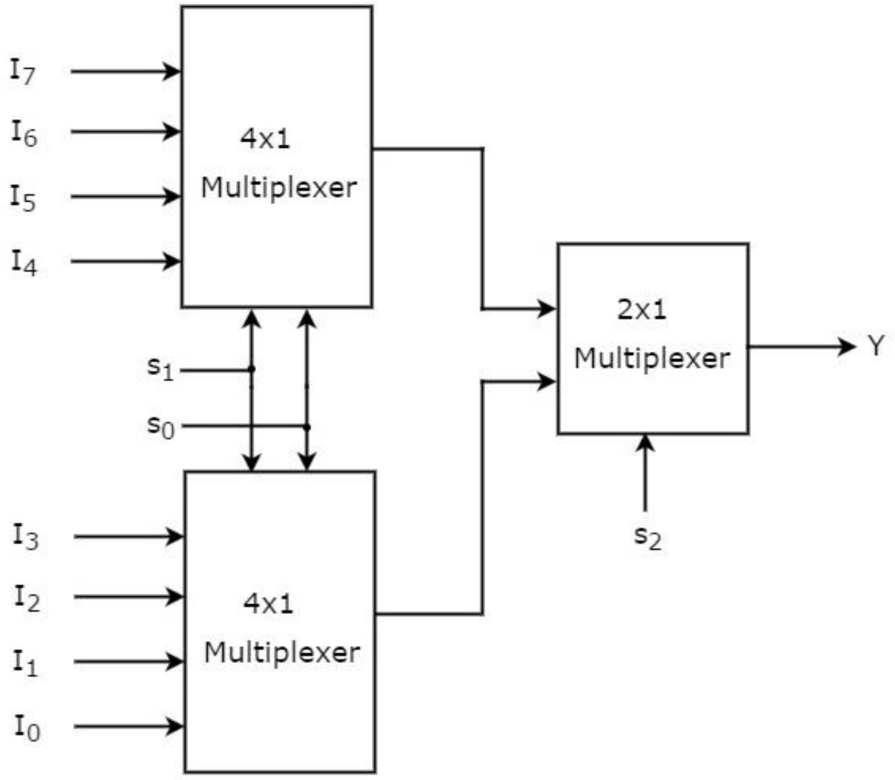
So, we require two **4x1 Multiplexers** in first stage in order to get the 8 data inputs. Since, each 4x1 Multiplexer produces one output, we require a **2x1 Multiplexer** in second stage by considering the outputs of first stage as inputs and to produce the final output.

Let the 8x1 Multiplexer has eight data inputs I_7 to I_0, three selection lines s_2, s_1 & s0 and one output Y. The **Truth table** of 8x1 Multiplexer is shown below.



Selection Inputs			Output
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

We can implement 8x1 Multiplexer using lower order Multiplexers easily by considering the above Truth table. The **block diagram** of 8x1 Multiplexer is shown in the following figure.



The same **selection lines, s_1 & s_0** are applied to both 4x1 Multiplexers. The data inputs of upper 4x1 Multiplexer are I_7 to I_4 and the data inputs of lower 4x1 Multiplexer are I_3 to I_0. Therefore, each 4x1 Multiplexer produces an output based on the values of selection lines, s_1 & s_0.

The outputs of first stage 4x1 Multiplexers are applied as inputs of 2x1 Multiplexer that is present in second stage. The other **selection line, s_2** is applied to 2x1 Multiplexer.

- If s_2 is zero, then the output of 2x1 Multiplexer will be one of the 4 inputs I_3 to I_0 based on the values of selection lines s_1 & s_0.
- If s_2 is one, then the output of 2x1 Multiplexer will be one of the 4 inputs I_7 to I_4 based on the values of selection lines s_1 & s_0.

Therefore, the overall combination of two 4x1 Multiplexers and one 2x1 Multiplexer performs as one 8x1 Multiplexer.

Function Table:

Enable	Selection Inputs		Outputs			
G	**B**	**A**	**Y0**	**Y1**	**Y2**	**Y3**
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

Table : 2*1 Multiplexer

H= Logic High, L= Logic Low, X= Don't Care

Designing an AND Gate using 2:1 MUX:

To design an AND using 2:1 mux, we need to tie the “zeroth” input to “Logic 0” and the “First” input to the one of the input of the AND Gate. The other input of AND gate would be connected with the select line of the MUX.

Now, the output of the MUX would be “1” only if the both of the inputs are “1” otherwise it would be “0” for all conditions.

Logic Diagram 2:1 MUX as an AND gate

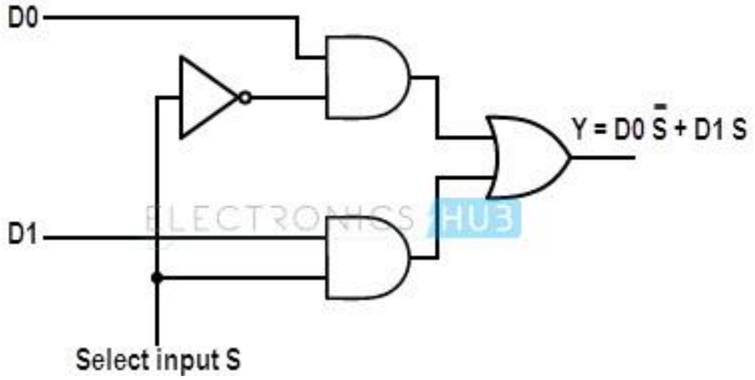


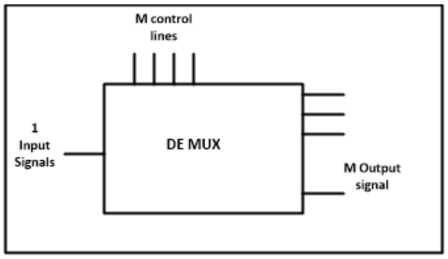
Fig 2: 2*1 Multiplexer using AND Gate

- An example of 4-to-1 multiplexer is IC 74153 in which the output is same as the input.
- Another example of 4-to-1 multiplexer is 45352 in which the output is the compliment of the input.
- Example of 16-to-1 line multiplexer is IC 74150.

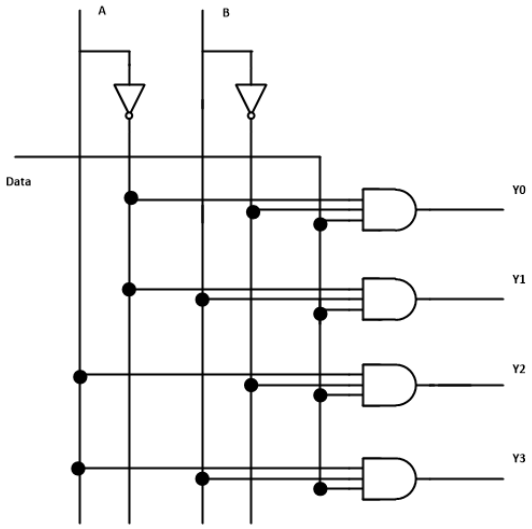
Demultiplexer

Demultiplexer means one to many. A demultiplexer is a circuit with one input and many outputs. 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.



Understanding 1-to-4 Demultiplexer
The 1-to-4 demultiplexer has 1 input bit, 2 control or select bits, and 4 output bits. An example of 1-to-4 demultiplexer is IC 74155. The 1-to-4 demultiplexer is shown in figure below-



The input bit is labelled as Data D. This data bit is transmitted to the selected output lines, which depends on the values of A and B, the control or Select Inputs.

- When A B = 0 1 , the second AND gate from the top is enabled while other AND gates are disabled. Therefore, data bit D is transmitted to the output Y1, giving Y1 = 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 A B = 1 0 , all the gates are disabled except the third AND gate from the top. Then, D is transmitted only to the Y2 output, and Y2 = Data.

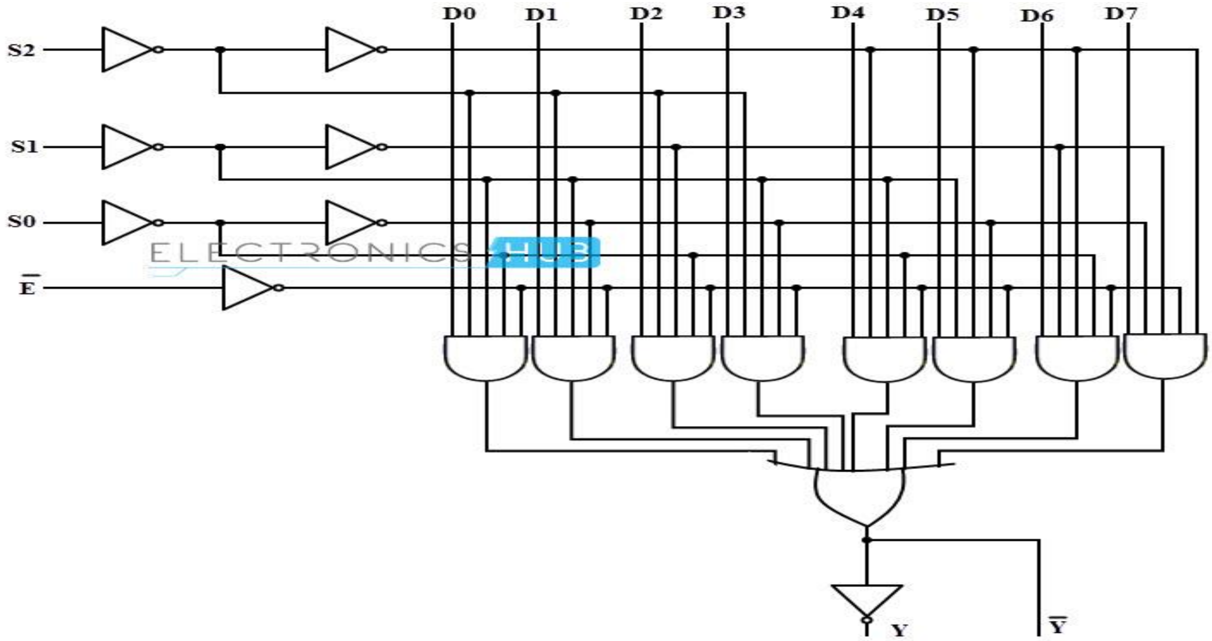
Example of 1-to-16 demultiplexer is IC 74154. It has 1 input bit, 4 control / select bits and 16 output bit.

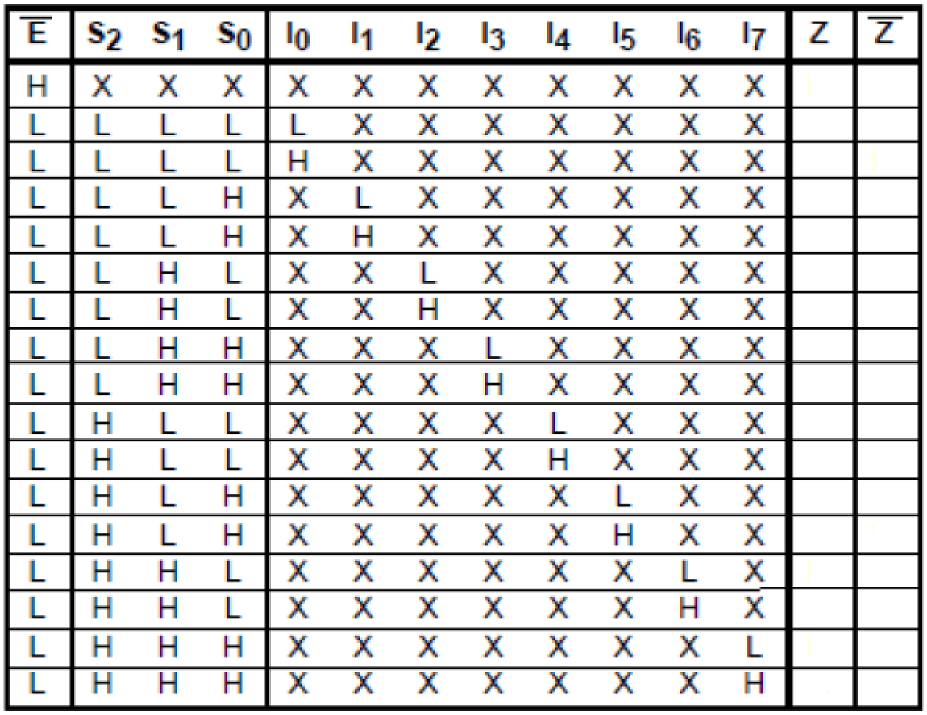
Applications of Demultiplexer

1. Demultiplexer is used to connect a single source to multiple destinations. The main application area of demultiplexer is communication system, where multiplexers 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 makes 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 attached 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.

Lab Task #1

Fill up the Truth Table below given the enable input, select inputs and the multiplexer





$\overline{E}$	S_2	S_1	S_0	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Z	$\overline{Z}$
H	X	X	X	X	X	X	X	X	X	X	X		
L	L	L	L	L	X	X	X	X	X	X	X		
L	L	L	L	H	X	X	X	X	X	X	X		
L	L	L	H	X	L	X	X	X	X	X	X		
L	L	L	H	X	H	X	X	X	X	X	X		
L	L	H	L	X	X	L	X	X	X	X	X		
L	L	H	L	X	X	H	X	X	X	X	X		
L	L	H	H	X	X	X	L	X	X	X	X		
L	L	H	H	X	X	X	H	X	X	X	X		
L	H	L	L	X	X	X	X	L	X	X	X		
L	H	L	L	X	X	X	X	H	X	X	X		
L	H	L	H	X	X	X	X	X	L	X	X		
L	H	L	H	X	X	X	X	X	H	X	X		
L	H	H	L	X	X	X	X	X	X	L	X		
L	H	H	L	X	X	X	X	X	X	H	X		
L	H	H	H	X	X	X	X	X	X	X	L		
L	H	H	H	X	X	X	X	X	X	X	H		

Lab Task #2

(I) Design NAND Logic Gate using 2:1 MUX.

(II) Design NOR Logic Gate using 2:1 MUX.

(III) Design XOR Logic Gate using 2:1 MUX.