AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH
Faculty of Engineering

Laboratory Report Cover Sheet
Students must complete all details except the faculty use part.

Please submit all reports to your subject supervisor or the office of the concerned faculty.

Lab Title: **Designing Multiplexer (MUX) and Demultiplexer (DEMUX), Encoder and Decoder Circuits.**

Experiment Number: **04** Due Date: 25/10/2023 Semester: Fall 2022-2023

Subject Code: **EEE3102** Subject Name: DIGITAL LOGIC AND CIRCUITS LAB Section: **L**

Course Instructor: **NUZAT NUARY ALAM** Degree Program: **B.Sc. CSE**

Declaration and Statement of Authorship:
1. I/we hold a copy of this report, which can be produced if the original is lost/ damaged.
2. This report is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this report has been written for me/us by any other person except where such collaboration has been authorized by the lecturer/teacher concerned and is clearly acknowledged in the report.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the school for review and comparison, including review by external examiners.

I/we understand that
7. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. Enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy your work.

Group Number (if applicable): 6 Individual Submission Group Submission

No.	Student Name	Student Number	Student Signature	Date
Submitted by:				
1	**MD. SHOHANUR RAHMAN SHOHAN**	**22-46013-1**		
Group Members:				
2	**RUDRO SHINE DATTA**	**22-46723-1**		
3	**A. H. M. TANVIR**	**22-47034-1**		
4	**ABIR BOKHTIAR**	**22-47038-1**		
5	**A. F. M. RAFIUL HASSAN**	**22-47048-1**		

For faculty use only:

Total Marks: _____ Marks Obtained: _____

Faculty comments_____

Title: Designing Multiplexer (MUX) and Demultiplexer (DEMUX), Encoder and Decoder Circuits.

Introduction:

In this experiment students will learn how to design and implement multiplexers (MUX) and demultiplexers (DeMUX) of different sizes using basic logic gates. They will also learn how to construct bigger multiplexers using smaller multiplexers. Students will also construct encoder and decoder circuits. Encoder and decoder circuits are very useful in information transmission, conversion, compression and maintaining the secrecy of any information.

Theory and Methodology:

Part I: Multiplexer and Demultiplexer

A multiplexer (or mux) is a device that selects one of several inputs and forwards the selected input into a single line. A multiplexer of 2n inputs has n selection lines, which are used to select which input must be sent to the output. A multiplexer is also called a data selector.

A demultiplexer (or demux) is a device taking a single input and selecting one of many data-output-lines, which is connected to the single input.

Multiplexer:

In a computer system, it is often necessary to choose data from exactly one of a number of possible sources. Suppose that there are four sources of data, provided as input signals D_0, D_1, D_2 and D_3. The values of these signals change in time, perhaps at regular intervals. We want to design a circuit that produces an output that has the same value as either D_0 or D_1 or D_2 or D_3, dependent on the values of two selection pins S_1 and S_0. Here, the number of selection pins is two. Four combinations are possible using these two selection pins S1 and S0, such as (S_1, S_0) = (0,0), (0,1), (1,0), (1,1). Each combination is dedicated for each input. Let us consider the output variable is f. Now if S1 = 0 and S_0 = 0 then f = D_0, if S_1 = 0 and S_0 = 1 then f = D_1, if S_1 = 1 and S_0 = 0 then f = D_2 and if S_1 = 1 and S_0 = 1 then f = D_3.

It is important to know that there is a relationship between the number of input and the number of selection pins. If the number of selection pin of a MUX is n, then maximum 2n inputs are possible for that MUX. And the MUX will be called 2nto1 line MUX. The MUX we are going to design is a 4to1 MUX. There could also be 2to1 MUX, 8to1 MUX, 16to1 MUX etc.

For our design, there are 4 inputs and 2 selection pins. So, we have 6 inputs. Now if we draw the truth table for 6 different inputs, there will be 64 input combinations. But fortunately, we can do it in a more convenient way as given below.

Table:1

S_1	S_0	f
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

From the above truth table, we can write the function as given below.

$$f = \bar{S}_1\bar{S}_0D_0+\bar{S}_1S_0D_1+S_1\bar{S}_0D_2+S_1S_0D_3\ldots (1)$$

The logic circuit of the equation (1) is given in figure 1.

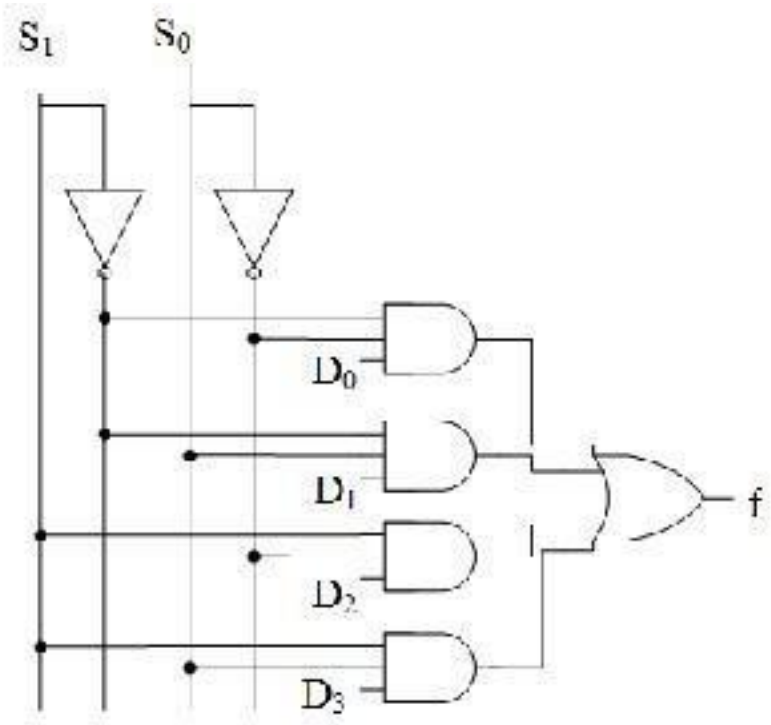


Figure1: 4to1 Multiplexer

Demultiplexer:

A Demultiplexer or Demux is opposite to the multiplexer. It has only one input and several outputs and one or more selection pins. Depending on the combination of selection input, the data input will be routed to one of many outputs. Other inputs will be low. Depending on the number of outputs, demultiplexers are termed as 1to2, 1to4 and 1to8 demultiplexers etc. If the number of selection pin is n, then maximum 2n outputs can be accommodated.

We are going to design a 1to4 line demux having an input Din, two selection pins S1 and S0 and four outputs D_0, D_1, D_2 and D_3. Now if $S_1 = 0$ and $S_0 = 0$ then $D_0 = D_{in}$, if $S_1 = 0$ and $S_0 = 1$ then $D_1 = D_{in}$, if $S_1 = 1$ and $S_0 = 0$ then $D_2 = D_{in}$ and if $S_1 = 1$ and $S_0 = 1$ then $D_3 = D_{in}$. We can draw the truth table as given below.

Table:2

S_1	S_0	D_0	D_1	D_2	D_3
0	0	D_{in}	0	0	0
0	1	0	D_{in}	D_{in}	0
1	0	0	0	0	0
1	1	0	0	0	D_{in}

From the above truth table, we can write the functions for D0, D1, D2 and D3 as given below.

$D_0 = \bar{S}_1\bar{S}_0D_{in}$ …… (2)
$D_1 = \bar{S}_1S_0D_{in}$ …… (3)
$D_2 = S_1\bar{S}_0D_{in}$ …… (4)
$D_3 = S_1S_0D_{in}$ …… (5)

The circuit for 1to4 line demux is given below.

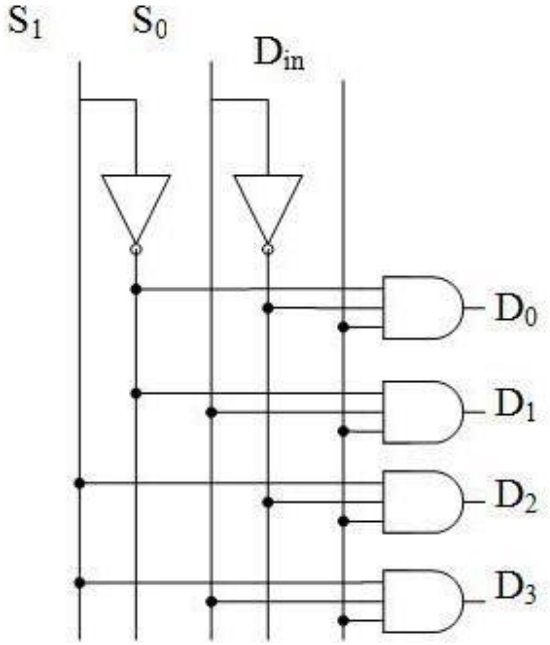


Figure 2: 1 to 4 Demultiplexer

It is also possible to construct 4to1 multiplexer (and 1to4 demultiplexer) using 2to1 multiplexers (1to2 demultiplexers) only. Figure 3 and figure 4 show the construction of 4to1 multiplexer using 2to1 multiplexers and 1to4 demultiplexer using 1to2 demultiplexers only.

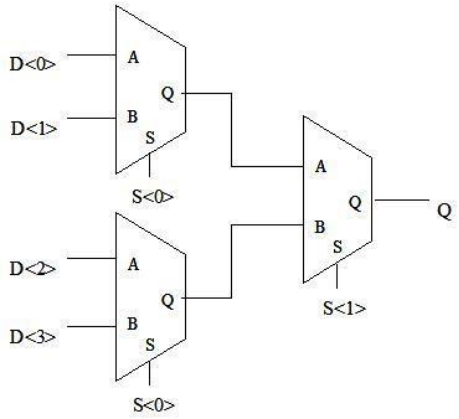


Figure 3: 4to1 multiplexer using 2to1 multiplexers.

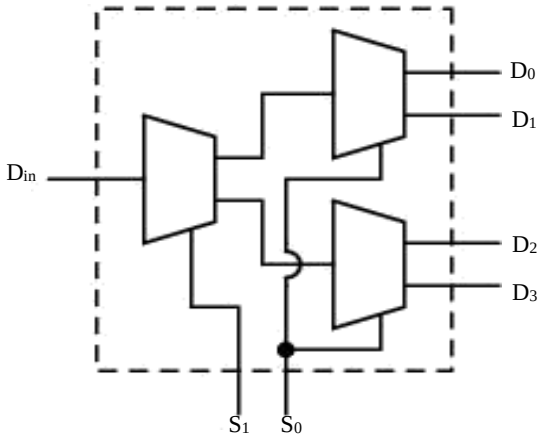


Figure 4: 1to4 demultiplexer using 1to2

Part II: Encoder and Decoder:

An encoder is a device or a circuit that converts information from one format or code to another. A decoder does the reverse operation of the encoder. It undoes the encoding so that the original information can be retrieved. Both the encoder and decoder are combinational circuits.

Encoding and decoding are very widely used ideas. They have applications in electronic circuits, software programs, medical devices, telecommunication, and many others. In this experiment, a very basic 2-to-4-line decoder and a decimal to BCD encoder will be constructed.

A decoder can convert binary information from n input lines to a maximum of 2n unique output lines. The 2-to-4-line decoder will take inputs from two lines and convert them to 4 lines.

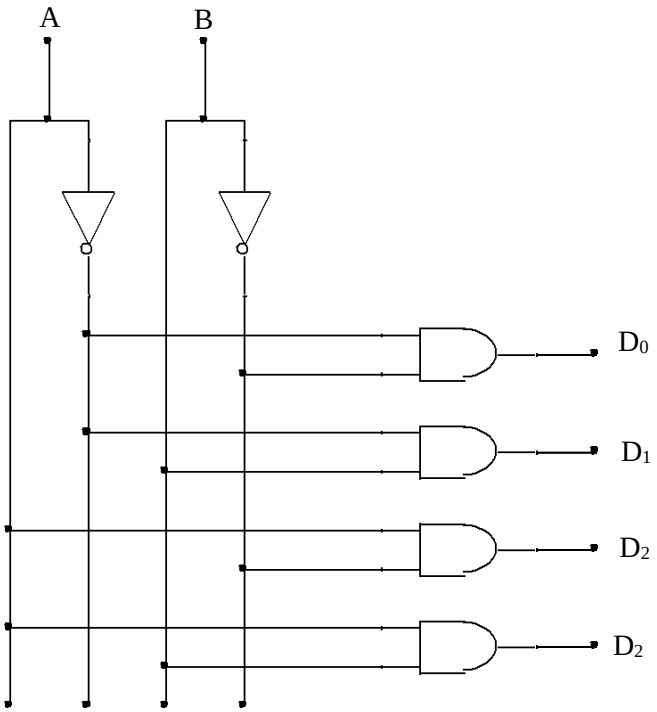


Fig.1: 2-to-4-line decoder

The expressions for implementing 2-to-4-line decoder –

$D_0 = \bar{A}\,\bar{B}$
$D_1 = \bar{A}B$
$D_2 = A\,\bar{B}$
$D_3 = AB$

Truth table for 2-to-4-line decoder is given below –

A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

A decimal to BCD encoder converts a decimal number into Binary Coded Decimal (BCD).

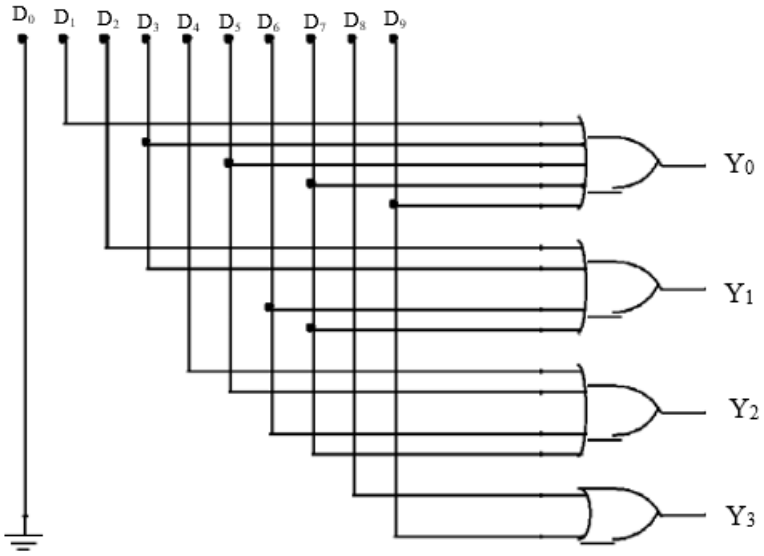


Fig.2: Decimal to BCD encoder

The expressions for implementing the decimal to BCD encoder –

$Y_0 = D_1 + D_3 + D_5 + D_7 + D_9$
$Y_1 = D_2 + D_3 + D_6 + D_7$
$Y_2 = D4 + D5 + D6 + D_7$
$Y_3 = D_8 + D_9$

Truth table for decimal to BCD encoder is given below –

Dec	Y_3	Y_2	Y_1	Y_0
D_0	0	0	0	0
D_1	0	0	0	1
D_2	0	0	1	0
D3	0	0	1	1
D_4	0	1	0	0
D_5	0	1	0	1
D_6	0	1	1	0
D_7	0	1	1	1
D_8	1	0	0	0
D_9	1	0	0	1

Priority encoder:

A priority encoder is a circuit or algorithm that compresses multiple binary inputs into a smaller number of outputs. The output of a priority encoder is the binary representation of the original number starting from zero of the most significant input bits. They are often used to control interrupt requests by acting on the highest priority request. If two or more inputs are given at the same time, the input having the highest priority will take precedence.

In this experiment a 4-to 2 priority encoders with a priority sequence of 2,1,3,0 has been shown. It means, in this priority encoder 2 has the highest priority and 0 has the lowest. If 2 is high then other

numbers are ignored (even if any of them are high at the same time) and output would be binary representation of 2, i.e., Y1Y0=10. If 2 is found to be low, then next priority is given to 1. So, in this case if 1 is high, then 3 and 0 are ignored and output will be binary representation of 1, i.e., Y1Y0=01 and so on.

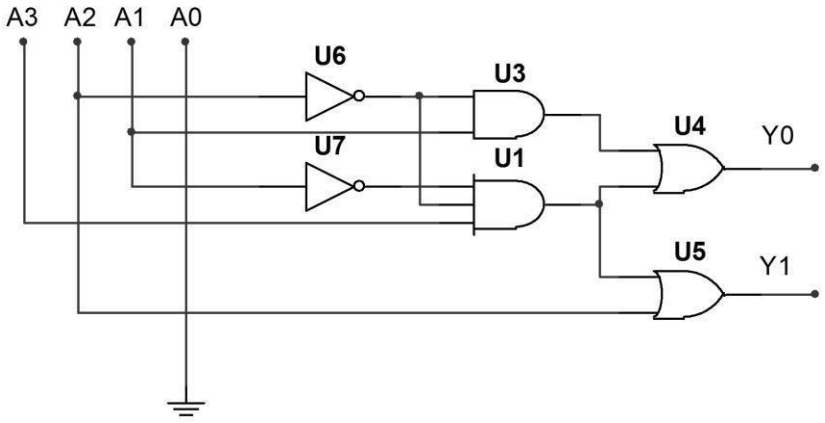


Fig .3: 4-to 2 priority encoders with a priority sequence of 2,1,3,0

The expressions for implementing the above priority encoder–

$Y_0 = \bar{A}_2 A_1 + A_3 \bar{A}_2 \bar{A}1$
$Y_1 = A_2 + A_3 \bar{A}_2 \bar{A}_1$

Truth table for this priority encoder is given below –

A_3	A_2	A_1	A_0	Y_1	Y_0
x	1	x	x	1	0
x	0	1	x	0	1
1	0	0	x	1	1
0	0	0	1	0	0

Pre-Lab Homework:

Read about the characteristics of encoder and decoder circuits from any book or websites and use PSIM to generate the output of the circuits provided in this lab sheet. Save the simulation results and bring it to the lab.

Apparatus:

NOT Gate-	IC 7404	1[pcs]
AND Gate-	IC 7408	1[pcs]
OR Gate-	5 input OR	1[pcs]
	4 input OR	2[pcs]
	2 input OR	1[pcs]

Precautions:

1. Make sure that all the LEDs and the toggle switches of the trainer board are working properly.
2. Do not shorten any connections. Short connection can produce heat (due to high current flow) which is harmful for the components.

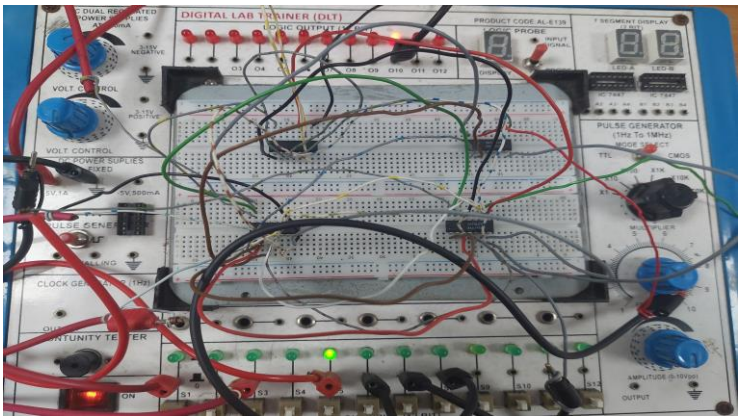
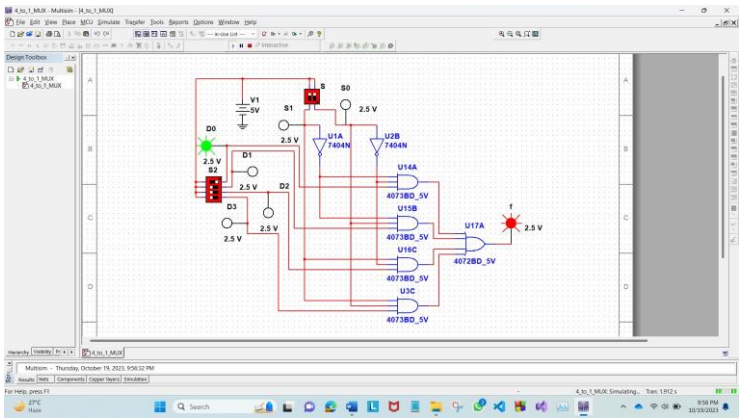
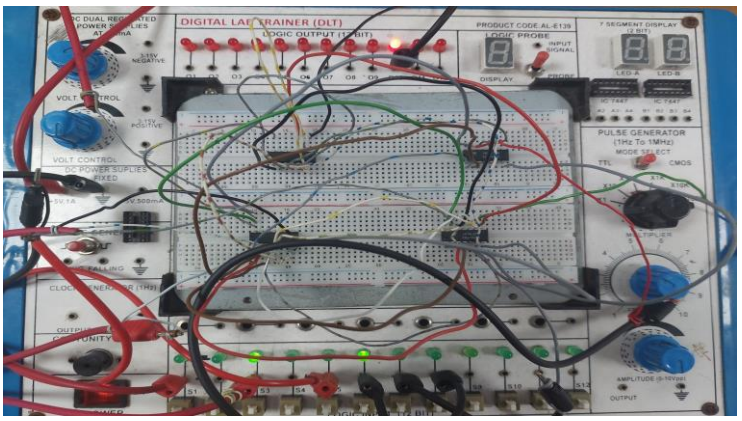
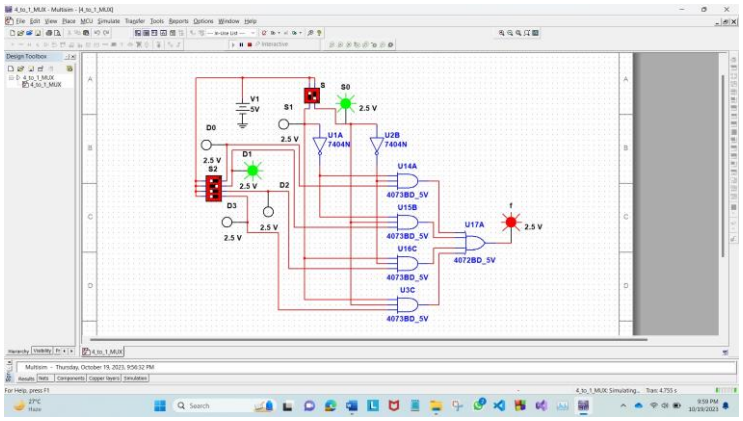
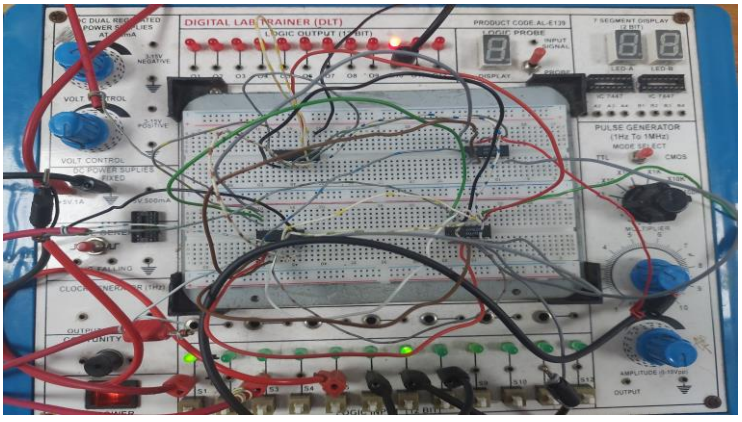
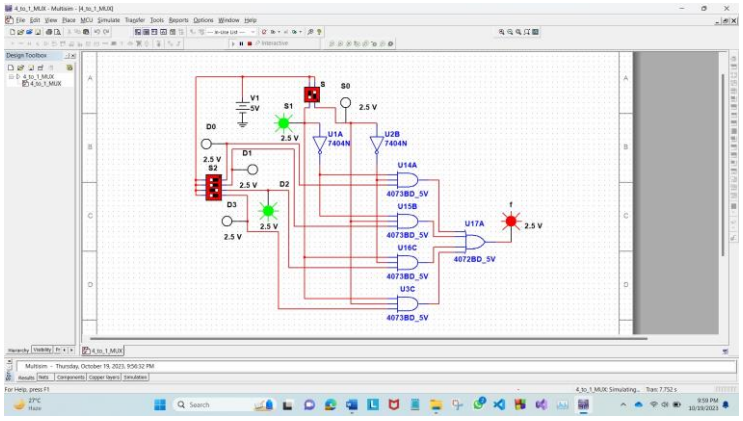
Experimental Procedure:

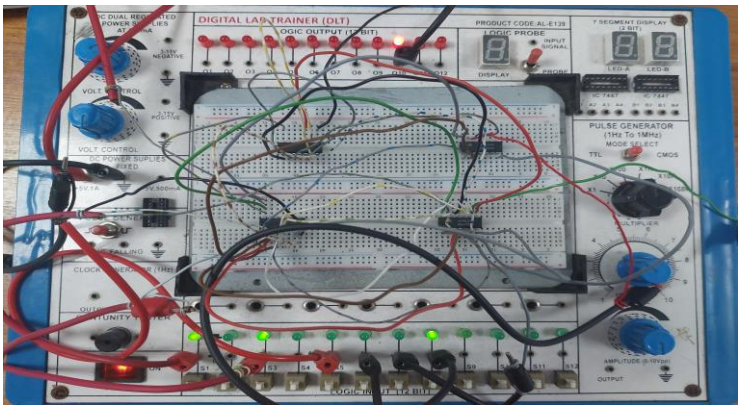
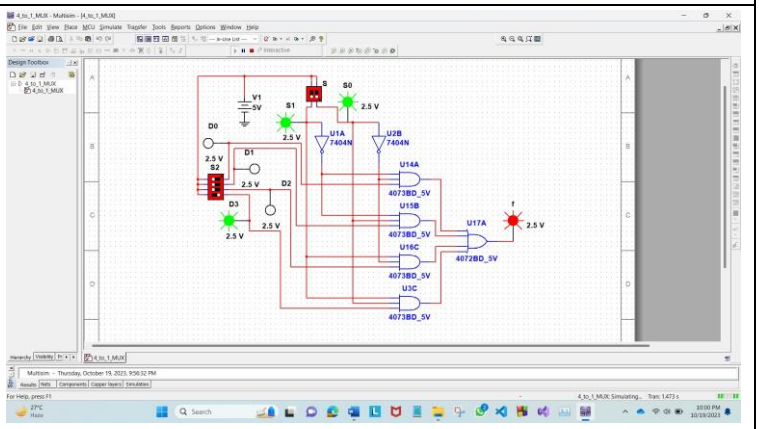
1. Connect the circuit according to the figures.
2. Use the toggle switches on the trainer board for providing input signal to the circuits. Connect the outputs to the LEDs on the trainer board.
3. Apply the input signals and observe and note the corresponding output signals.

Simulation and Measurement:

4to1 Multiplexer:

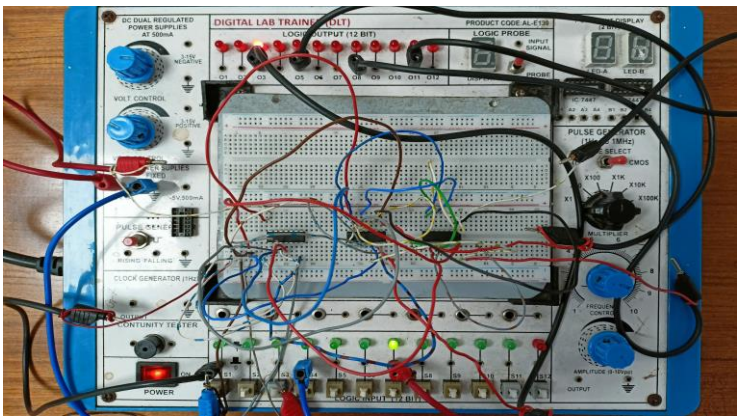
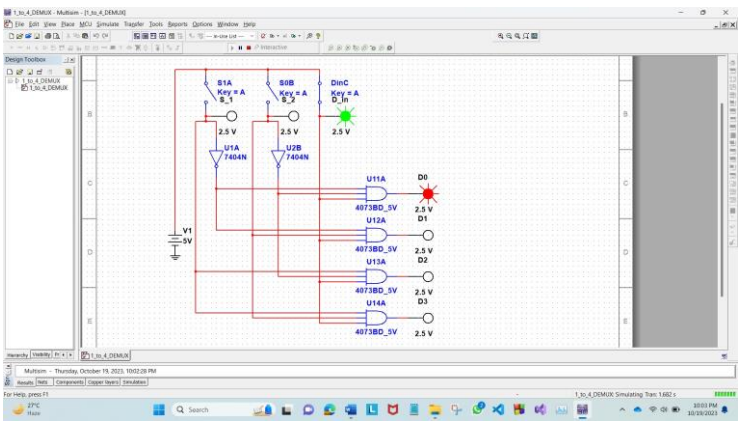
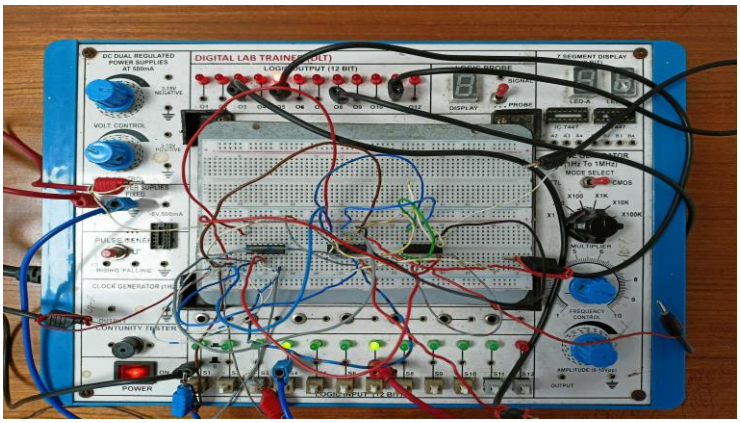
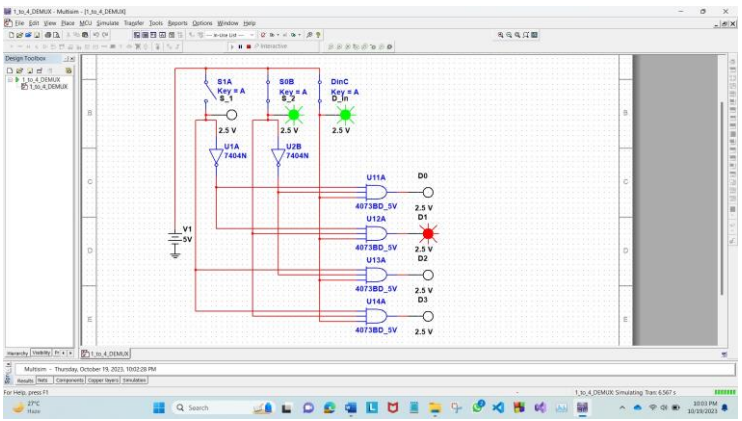
$$f = \bar{S}_1\bar{S}_0D_0 + \bar{S}_1S_0D_1 + S_1\bar{S}_0D_2 + S_1S_0D_3$$

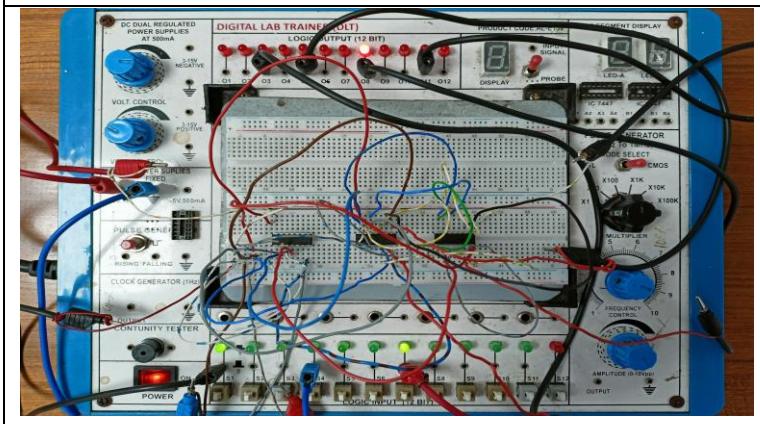
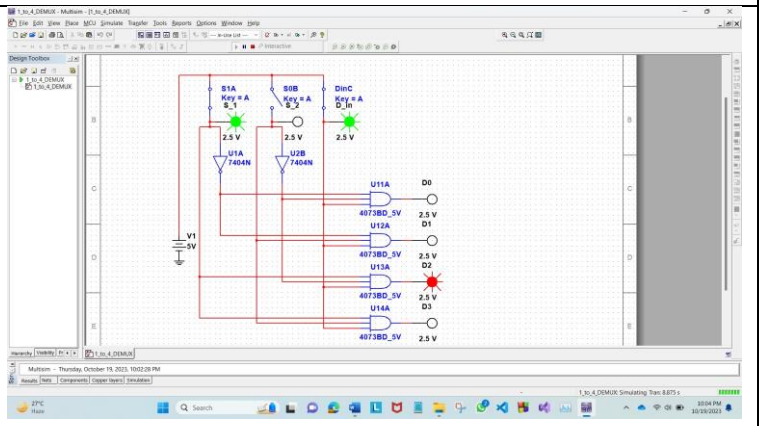
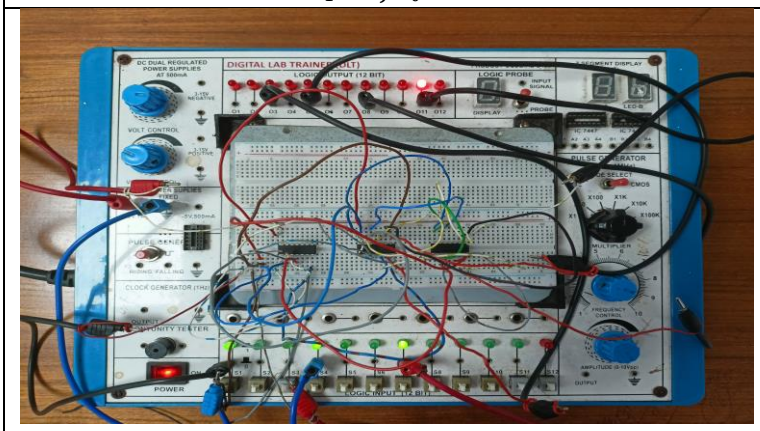
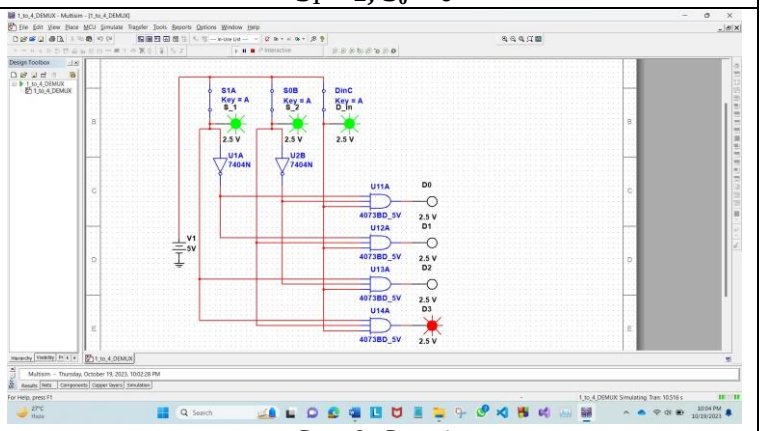
Hardware	Simulation
	
$S_1 = 0, S_0 = 0$ & $f = D_0$	**$S_1 = 0, S_0 = 0$ & $f = D_0$**
	
$S_1 = 0, S_0 = 1$ & $f = D_1$	**$S_1 = 0, S_0 = 1$ & $f = D_1$**
	
$S_1 = 1, S_0 = 0$ & $f = D_2$	**$S_1 = 1, S_0 = 0$ & $f = D_2$**

Hardware	Simulation
	
$S_1 = 1, S_0 = 1$ & $f = D_3$	$S_1 = 1, S_0 = 1$ & $f = D_3$

1 to 4 Demultiplexer:

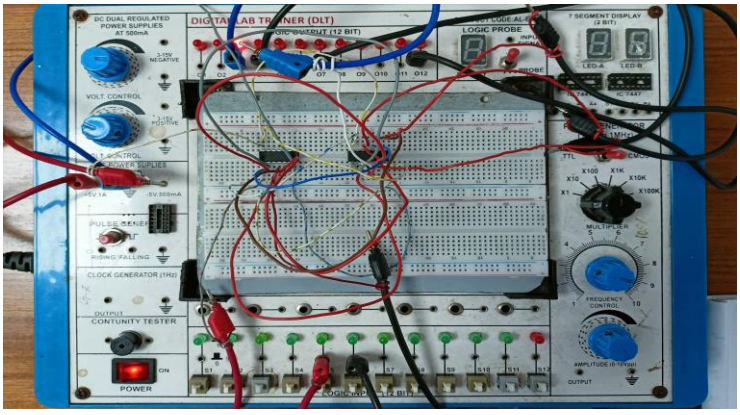
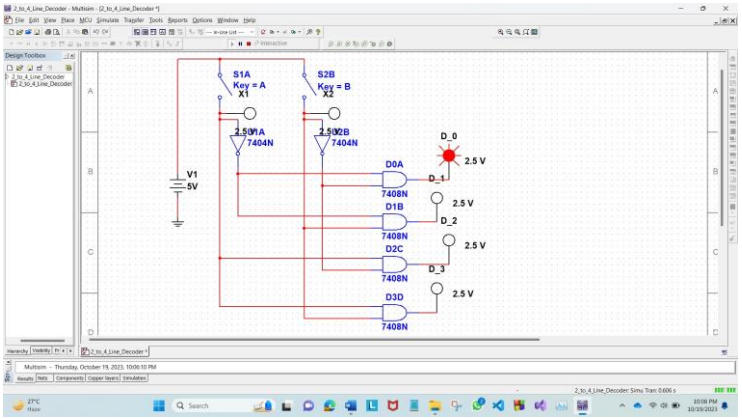
$$D_0 = \bar{S}_1\bar{S}_0D_{in}$$
$$D_1 = \bar{S}_1S_0D_{in}$$
$$D_2 = S_1\bar{S}_0D_{in}$$
$$D_3 = S_1S_0D_{in}$$

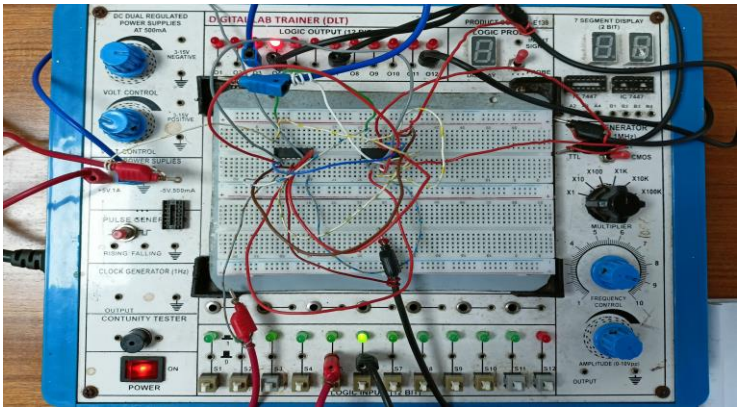
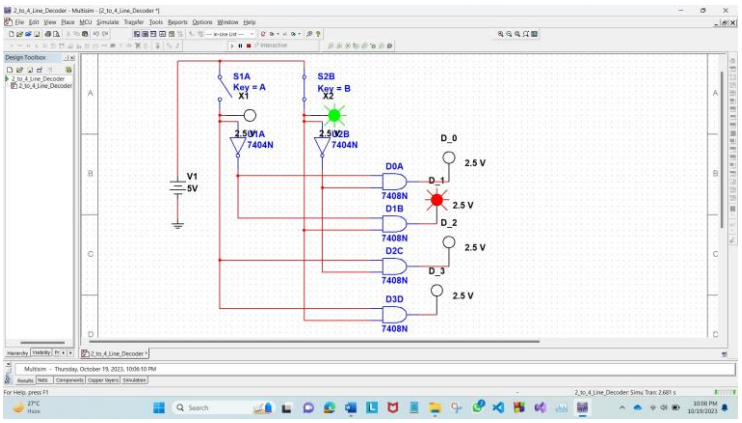
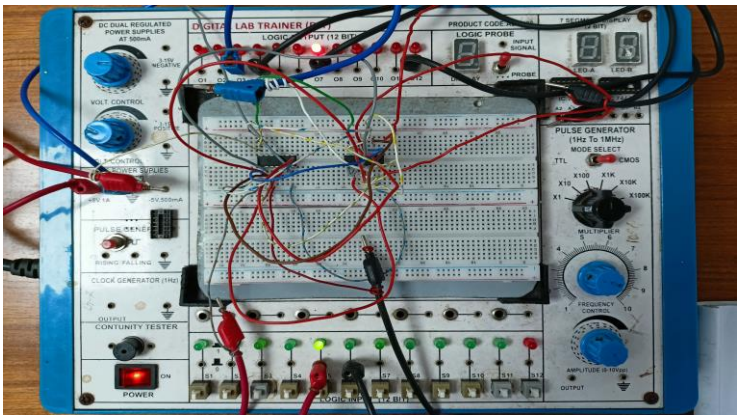
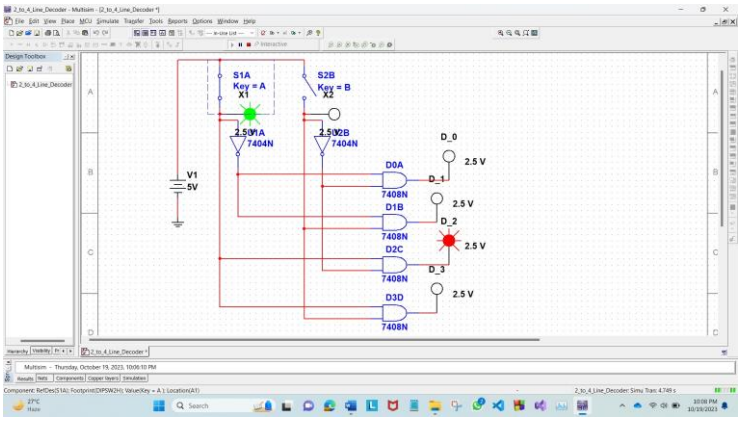
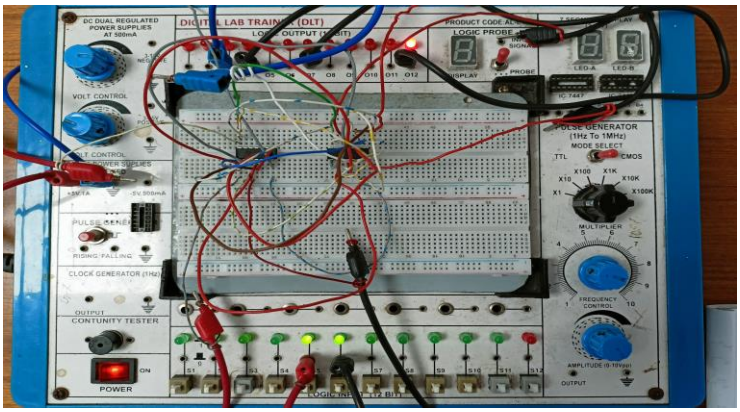
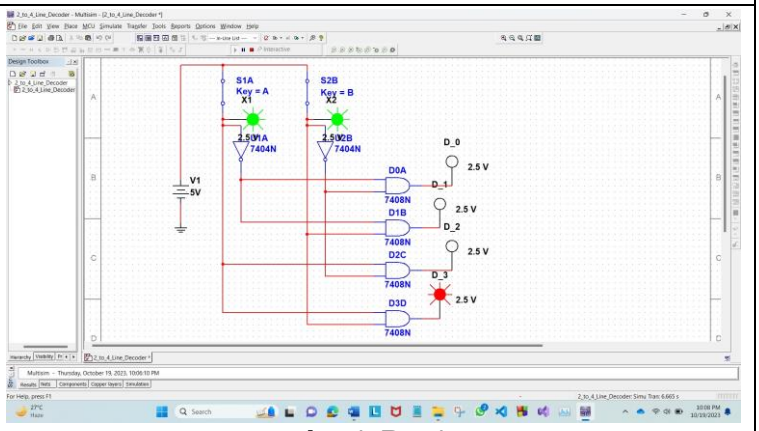
Hardware	Simulation
	
$S_1 = 0, S_0 = 0$	$S_1 = 0, S_0 = 0$
	
$S_1 = 0, S_0 = 1$	$S_1 = 0, S_0 = 1$

Hardware	Simulation
	
$S_1 = 1, S_0 = 0$	$S_1 = 1, S_0 = 0$
	
$S_1 = 0, S_0 = 1$	$S_1 = 0, S_0 = 1$

2-to-4-line Decoder:

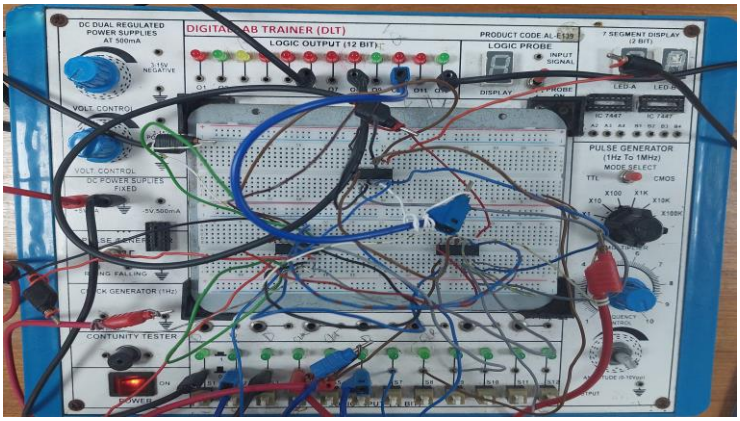
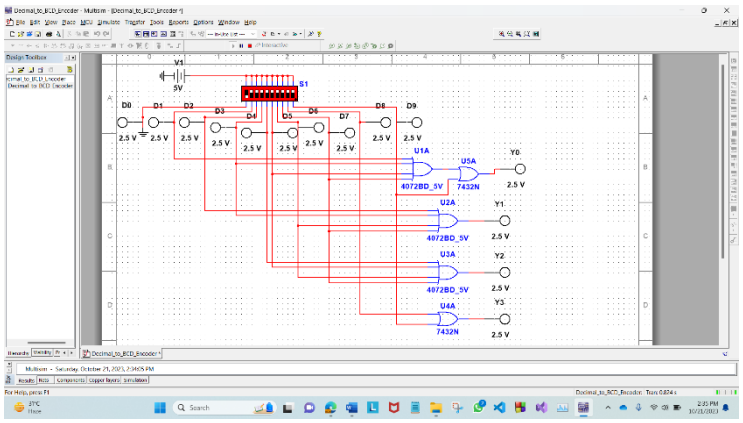
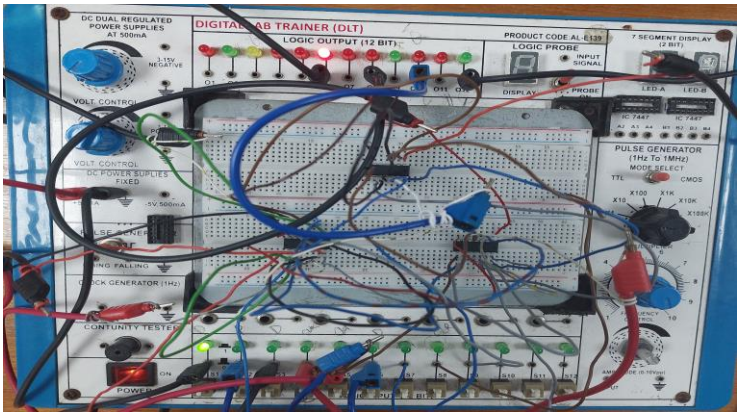
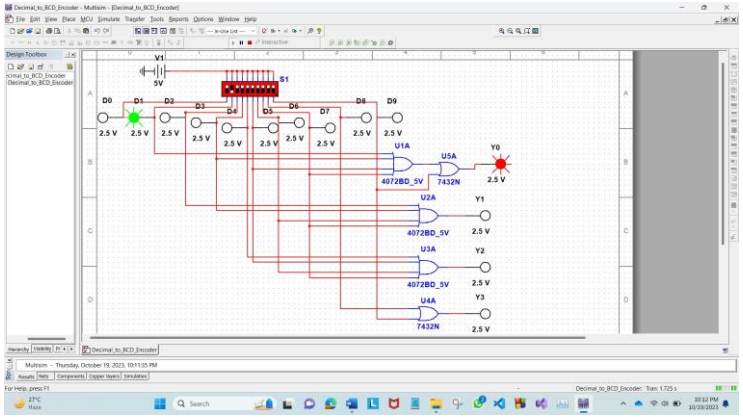
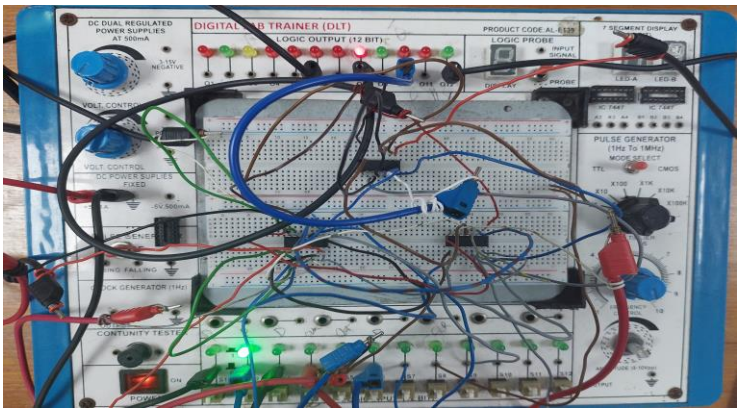
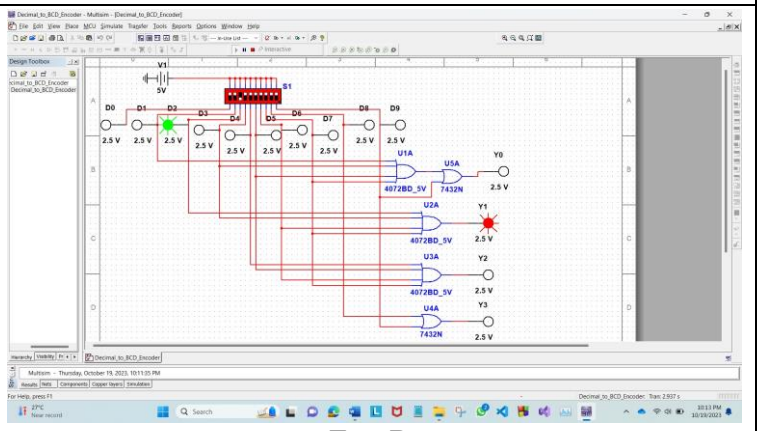
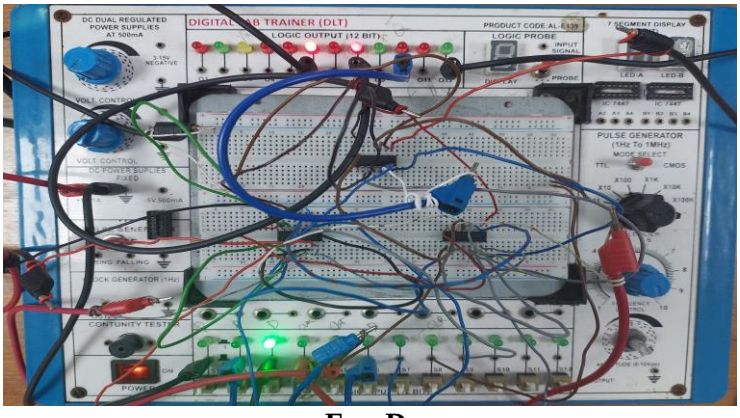
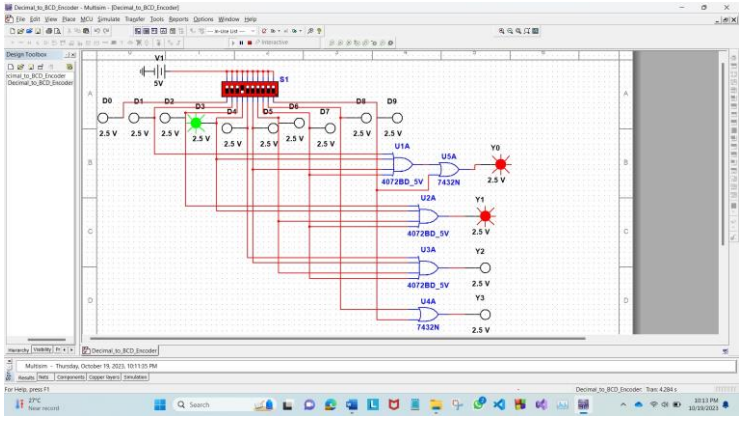
$D_0 = \bar{A}\,\bar{B}$
$D_1 = \bar{A}\,B$
$D_2 = A\,\bar{B}$
$D_3 = A\,B$

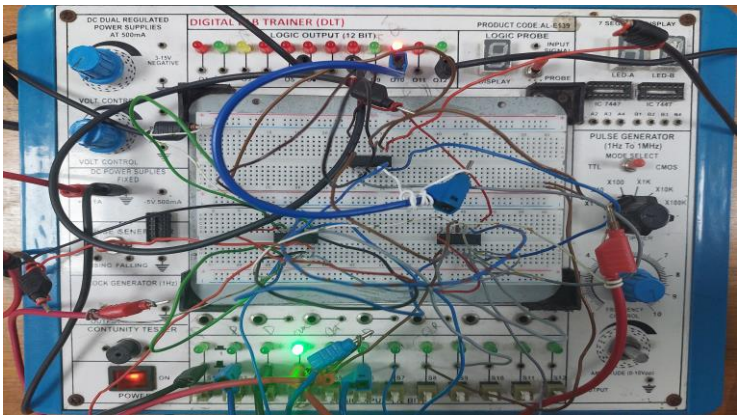
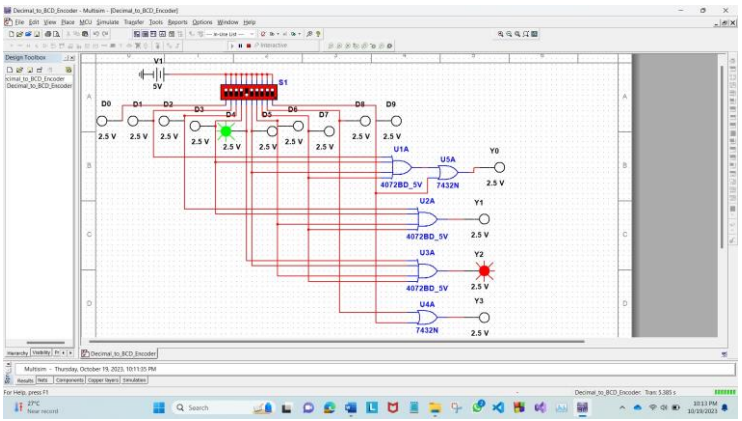
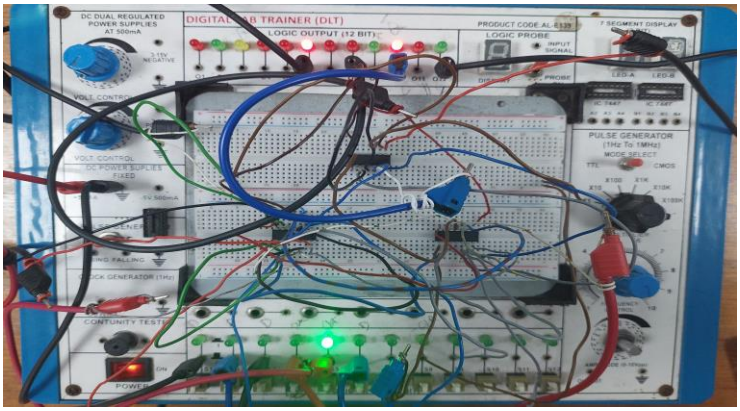
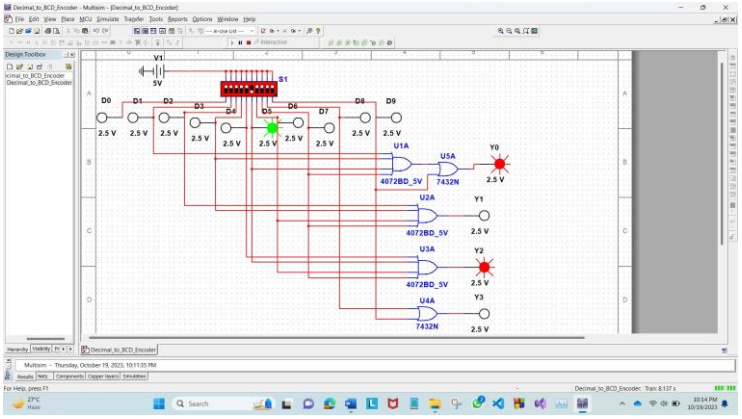
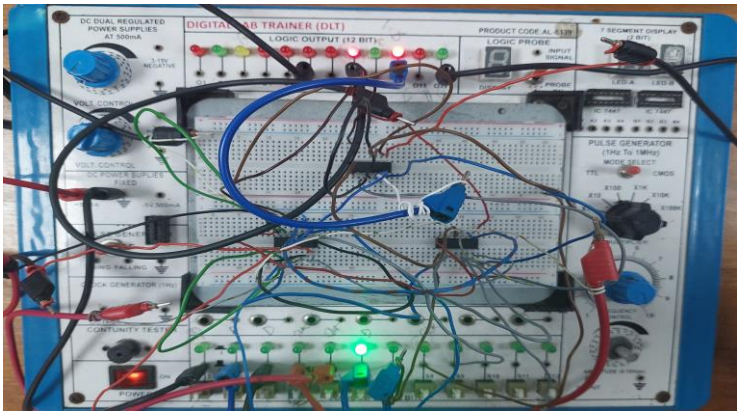
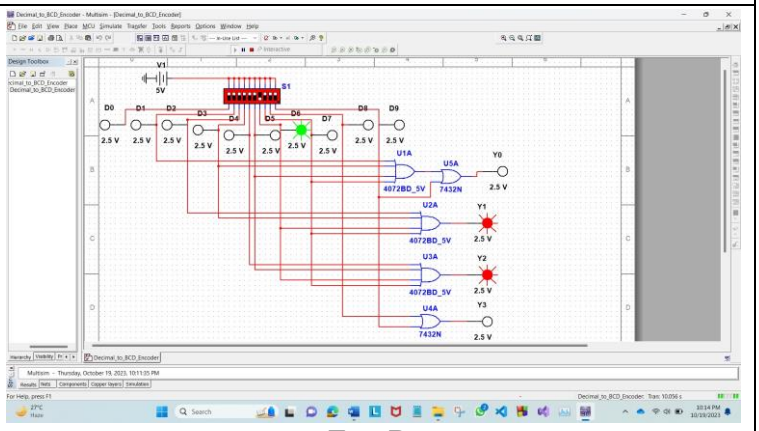
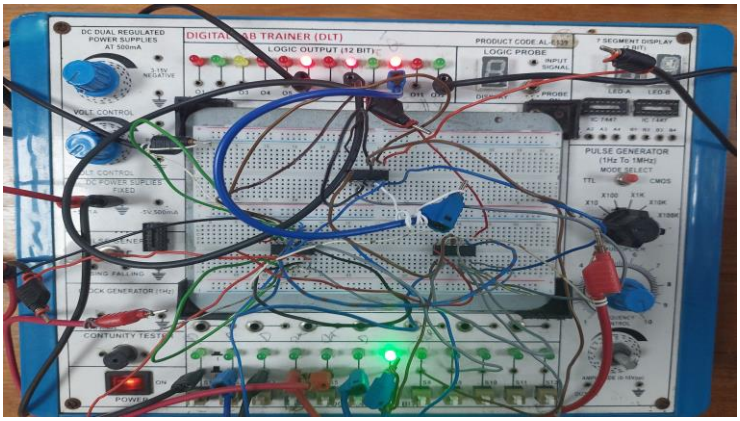
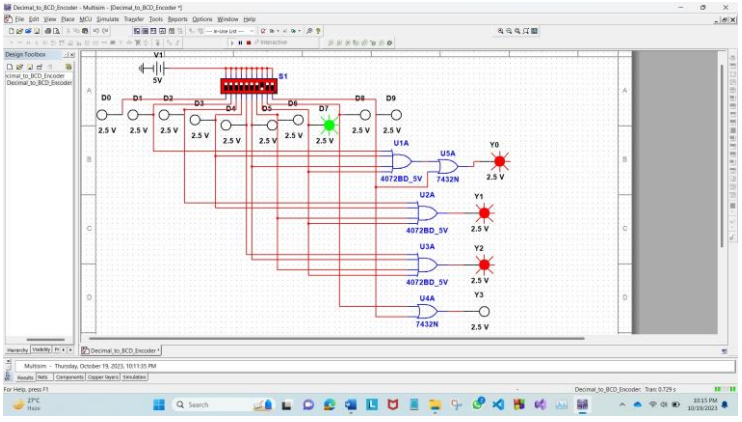
Hardware	Simulation
	
$A = 0, B = 0$	$A = 0, B = 0$

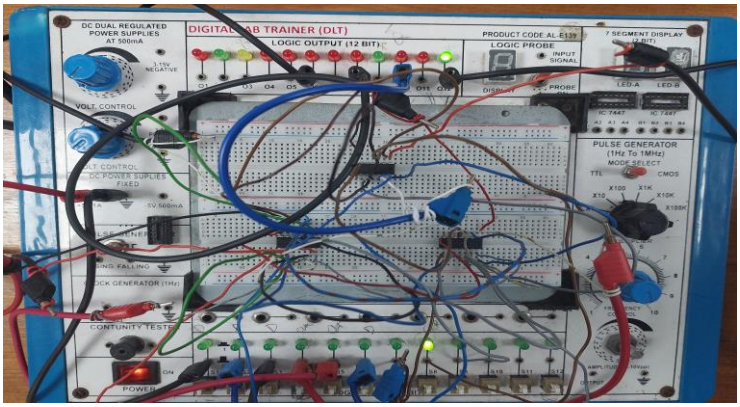
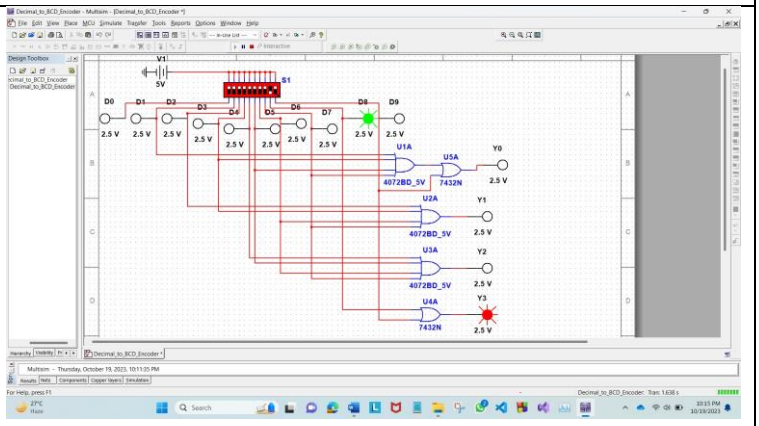
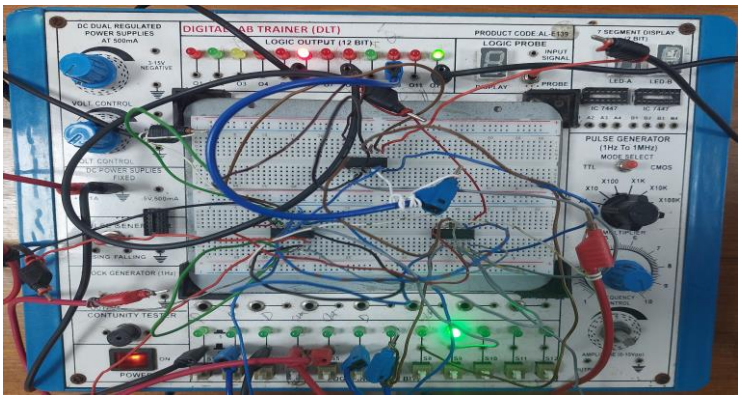
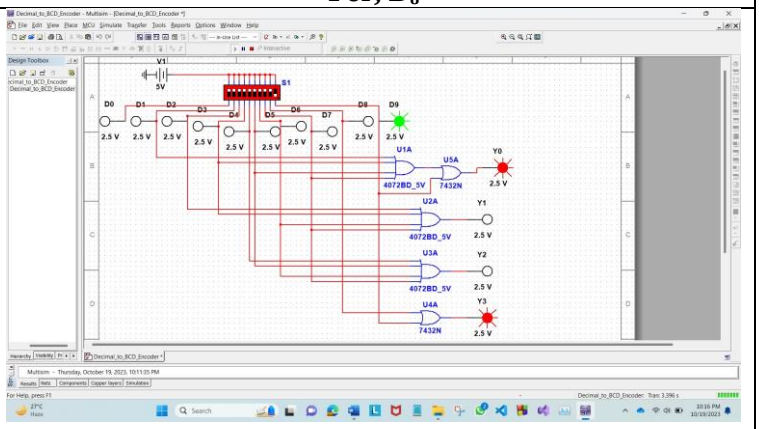
Hardware	Simulation
	
A = 0, B = 1	**A = 0, B = 1**
	
A = 1, B = 0	**A = 1, B = 0**
	
A = 1, B = 1	**A = 1, B = 1**

Decimal to BCD encoder:

$$Y_0 = D_1 + D_3 + D_5 + D_7 + D_9$$
$$Y_1 = D_2 + D_3 + D_6 + D_7$$
$$Y_2 = D4 + D5 + D6 + D_7$$
$$Y_3 = D_8 + D_9$$

Hardware	Simulation
	
For, D_0	For, D_0
	
For, D_1	For, D_1
	
For, D_2	For, D_2
	
For, D_3	For, D_3

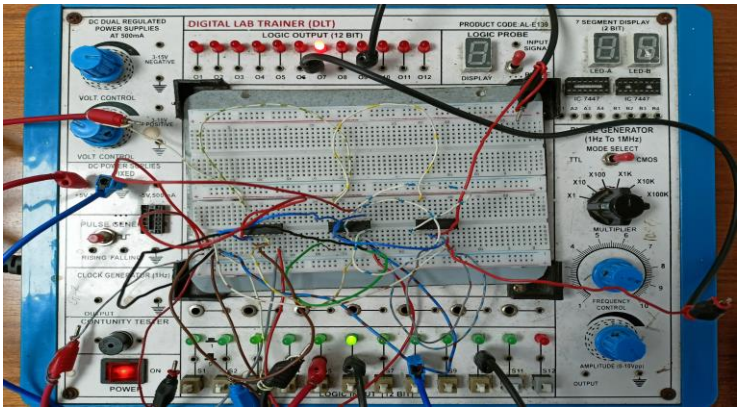
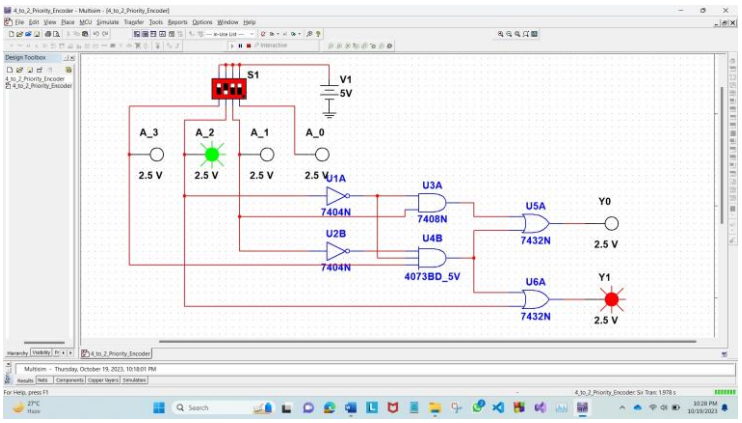
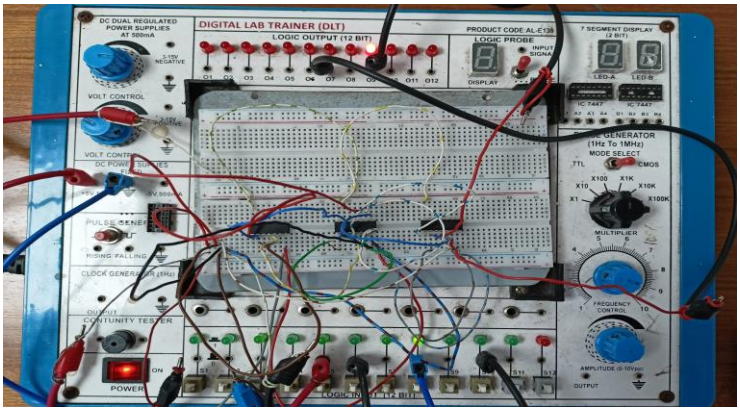
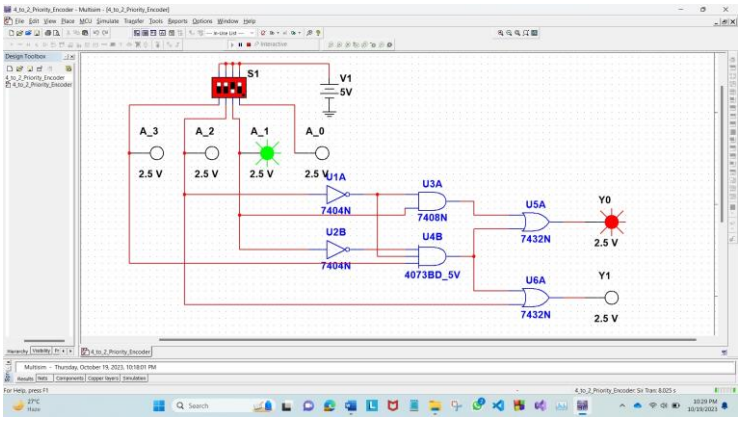
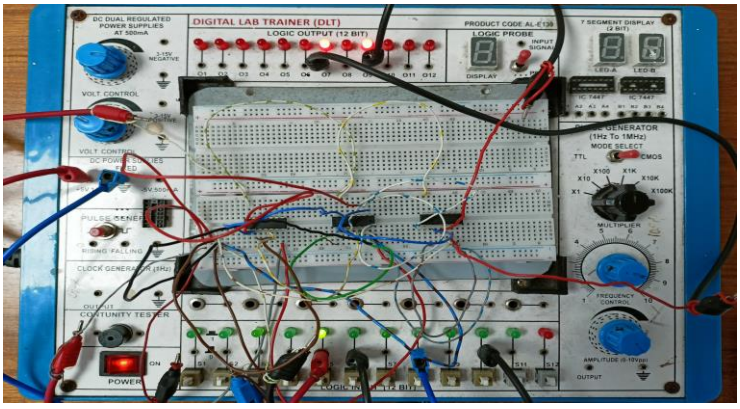
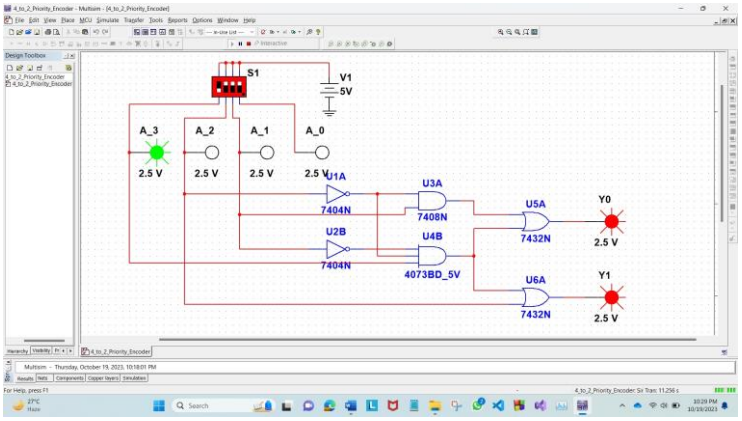
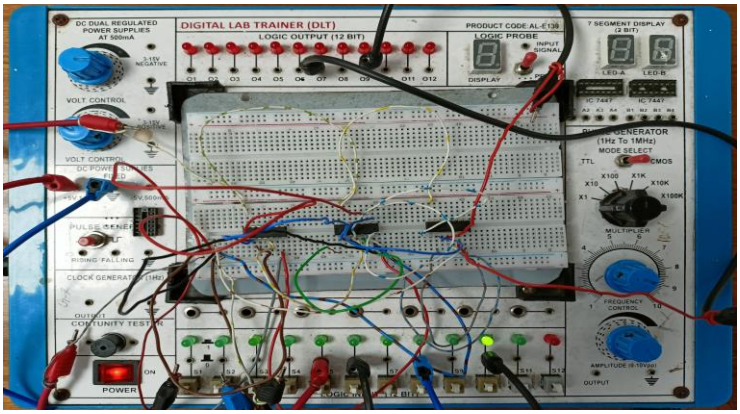
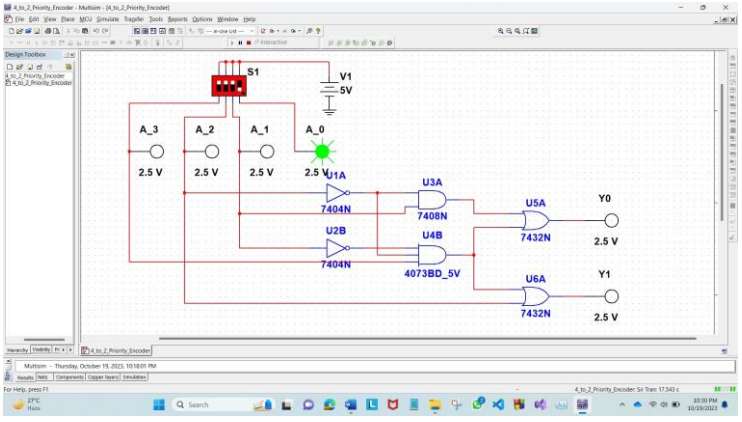
Hardware	Simulation
	
For, D_4	For, D_4
	
For, D_5	For, D_5
	
For, D_6	For, D_6
	
For, D_7	For, D_7

Hardware	Simulation
	
For, D_8	**For, D_8**
	
For, D_9	**For, D_9**

4-to 2 priority Encoder (priority sequence of 2,1,3,0):

$$Y_0 = \bar{A}_2 A_1 + A_3 \bar{A}_2 \bar{A}_1$$

$$Y_1 = A_2 + A_3 \bar{A}_2 \bar{A}_1$$

Hardware	Simulation
	
$A_3 = x, A_2 = 1, A_1 = 0, A_0 = x$	$A_3 = x, A_2 = 1, A_1 = 0, A_0 = x$
	
$A_3 = x, A_2 = 0, A_1 = 1, A_0 = x$	$A_3 = x, A_2 = 0, A_1 = 1, A_0 = x$
	
$A_3 = 1, A_2 = 0, A_1 = 0, A_0 = x$	$A_3 = 1, A_2 = 0, A_1 = 0, A_0 = x$
	
$A_3 = 0, A_2 = 0, A_1 = 0, A_0 = 1$	$A_3 = 0, A_2 = 0, A_1 = 0, A_0 = 1$

Results and Discussion:

In this experiment, we designed and implemented fundamental digital circuits including a 4 to 1 Multiplexer, 1 to 4 Demultiplexer, 2-to-4-line decoder, Decimal to BCD encoder, and a 4 to 2 priority encoder with a sequence of 2, 1, 3, 0. We successfully verified their functionality through hardware simulation using Multisim, confirming that these circuits perform as expected. This practical experience enhances our understanding of essential digital components, their applications in data routing, memory addressing, and control logic, and highlights the value of simulation tools in real-world circuit design. The knowledge gained here lays the foundation for more advanced digital circuit applications in future experiments.

References:

[1] http://www.tutorialspoint.com/computer_logical_organization/combinational_circuits.htm