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 multiplexer 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 2^n inputs has n selection lines, which are used to select which input has to 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 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 pin is two. Four combinations are possible using these two selection pins S_1 and S_0, 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 $S_1 = 0$ 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 2^n inputs are possible for that MUX. And the MUX will be called as 2^nto1 line MUX.

The MUX we are going to design is a 4to1 MUX. There could be also 2to1 MUX, 8to1 MUX, 16to1 MUX etc.

For our design, there are 4 inputs and 2 selection pins. So actually 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.

$$f = S_1\overline{S_0}D_0 + \overline{S_1}S_0D_1 + S_1S_0\overline{D_2} + S_1S_0D_3 \ldots (1)$$

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

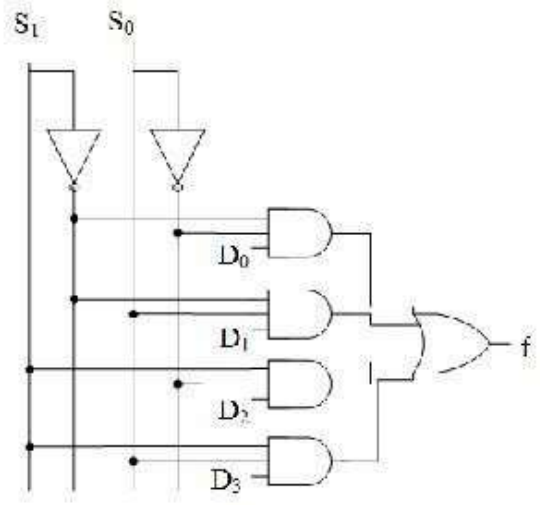


Figure1: 4to1 Multiplexer

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 output, demultiplexers are termed as 1to2, 1to4 and 1to8 demultiplexers etc. If the number of selection pin is n, then maximum 2^n outputs can be accommodated.

We are going to design a 1to4 line demux having an input D_{in}, two selection pins S_1 and S_0 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}	0	0
1	0	0	0	D_{in}	0
1	1	0	0	0	D_{in}

From the above truth table we can write the functions for D_0 ,D_1, D_2 and D_3 as given below.

$D_0 = S_1\bar{S}_0\bar{D}_{in}$ … (2)
$D_1 = S_1\bar{S}_0D_{in}$ … (3)
$D_2 = S_1S_0\bar{D}_{in}$ … (4)
$D_0 = S_1\bar{S}_0\bar{D}_{in}$ … (5)

The circuit for 1to4 line demux is given below.

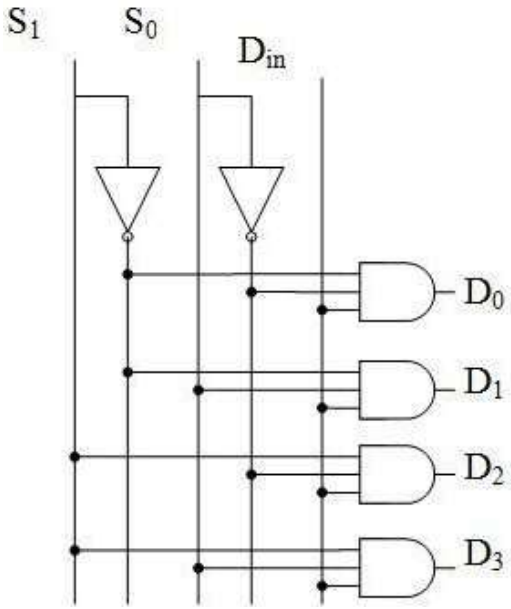


Figure 2: 1 to 4 Demultiplexer

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.

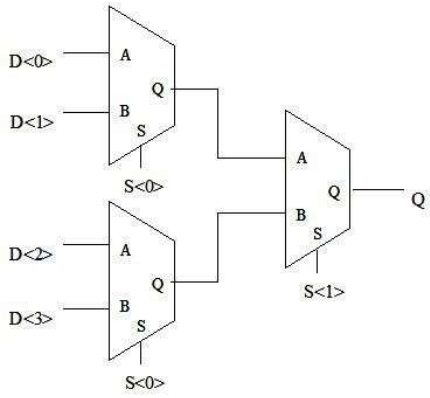


Figure 3: 4to1 multiplexer using 2to1 multiplexers.

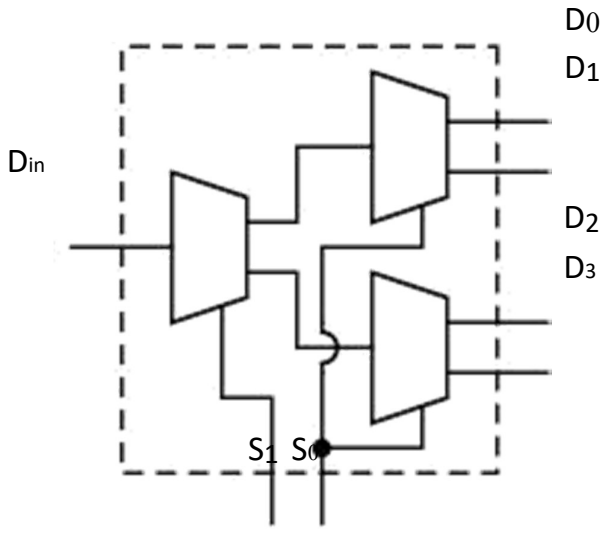


Figure 4: 1to4 demultiplexer using 1to2 demultiplexers.

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 2^n unique output lines. The 2-to-4 line decoder will take inputs from two lines and convert them to 4 lines.

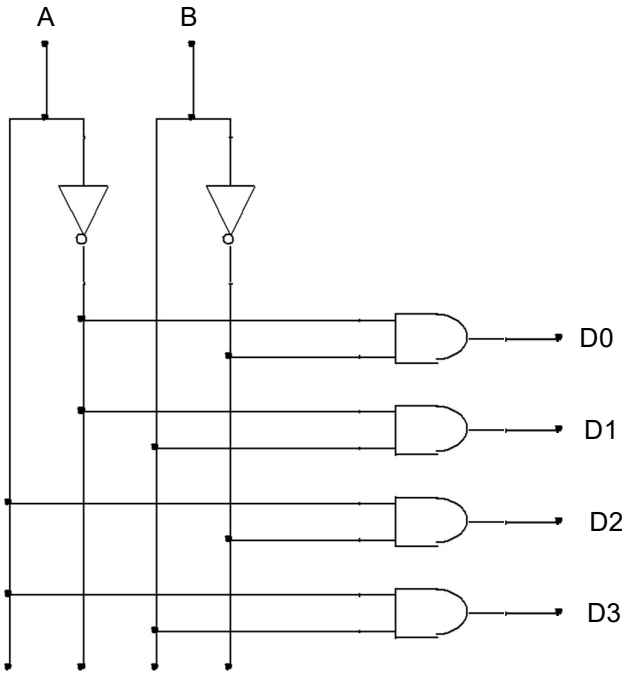


Fig.1: 2-to-4 line decoder

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

D0 = A’B’
D1 = A’B
D2 = AB’
D3 = AB

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

A	B	D0	D1	D2	D3
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).

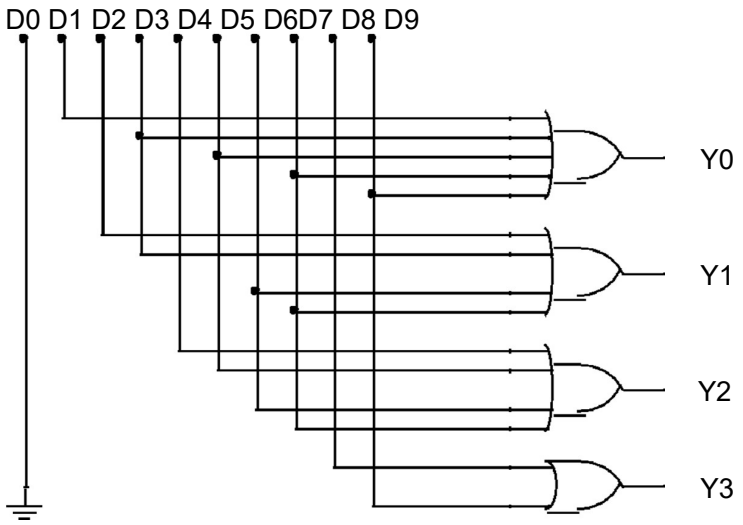


Fig.2: Decimal to BCD encoder

The expressions for implementing the decimal to BCD encoder –

Y0 = D1 + D3 + D5 + D7 + D9
Y1 = D2 + D3 + D6 + D7
Y2 = D4 + D5 + D6 + D7
Y3 = D8 + D9

Truth table for decimal to BCD encoder is given below –

Dec.	Y3	Y2	Y1	Y0
D0	0	0	0	0
D1	0	0	0	1
D2	0	0	1	0
D3	0	0	1	1
D4	0	1	0	0
D5	0	1	0	1
D6	0	1	1	0
D7	0	1	1	1
D8	1	0	0	0
D9	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 bit. 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 encoder 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., Y_1Y_0=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., Y_1Y_0=01 and so on.

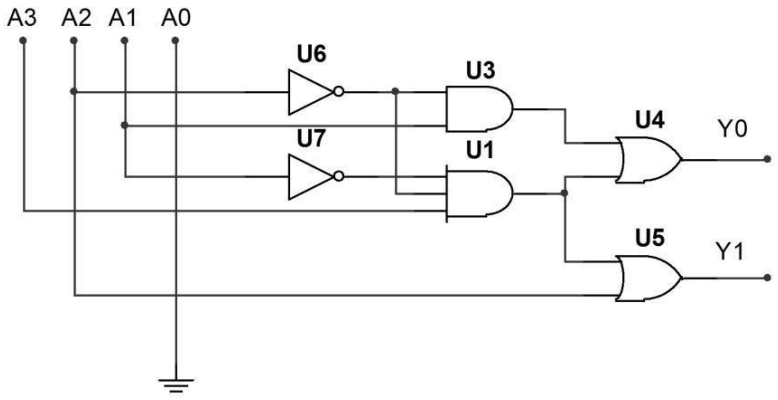


Fig .3: 4-to 2 priority encoder with a priority sequence of 2,1,3,0

The expressions for implementing the above priority encoder–

Y0 = A2'.A1 +
A3.A2'.A1' Y1
Y1= A2 +
A3.A2'.A1'

Truth table for this priority encoder is given below –

A3	A2	A1	A0	Y1	Y0
x	1	x	x	1	0
x	0	1	x	0	1
1	0	0	x	1	1
0	0	0	1	0	0

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]

Simulation and Measurement:

Part II

2-to-4 line decoder

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

A	B	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

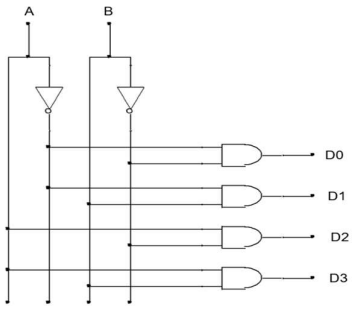
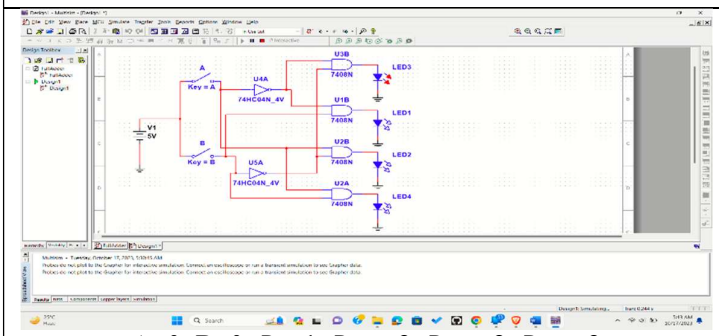
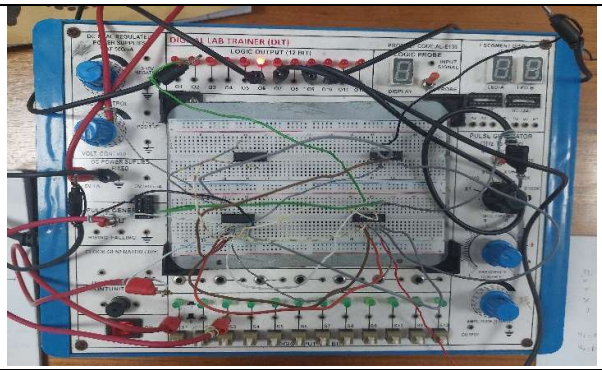
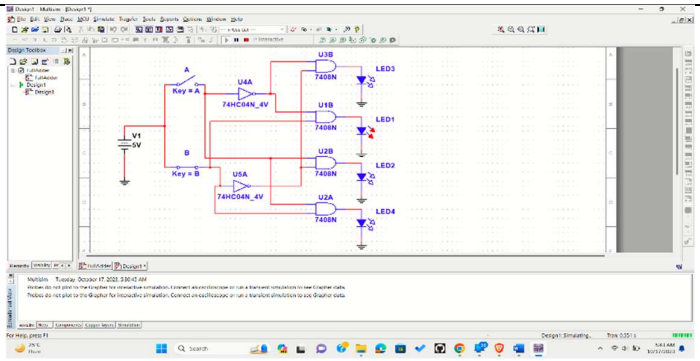
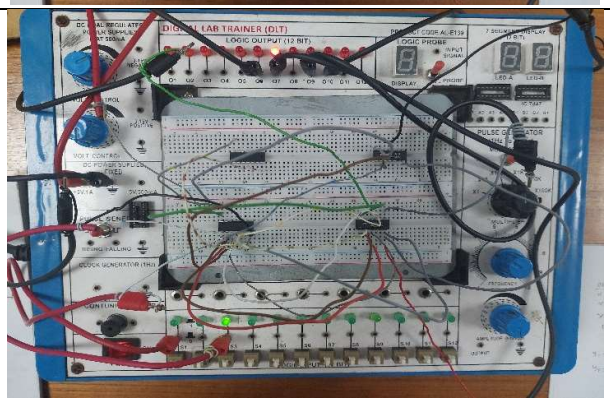
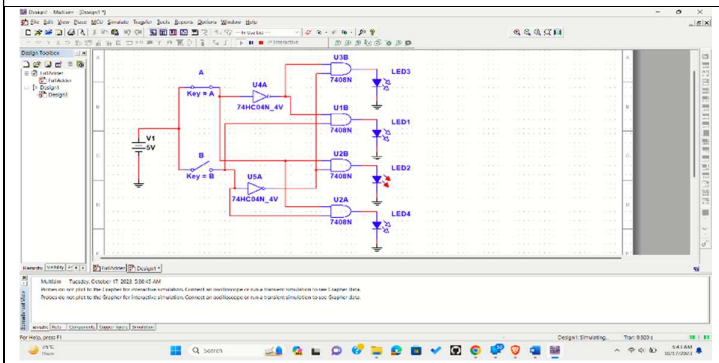
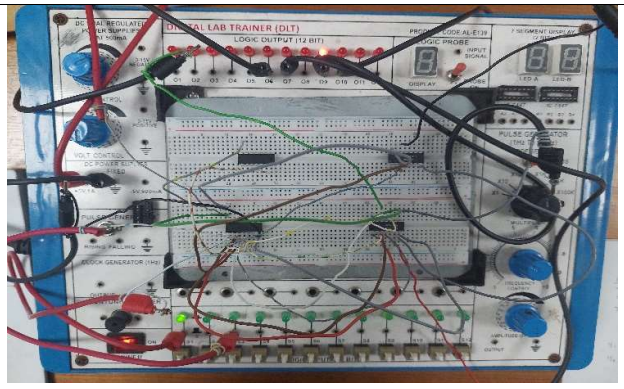
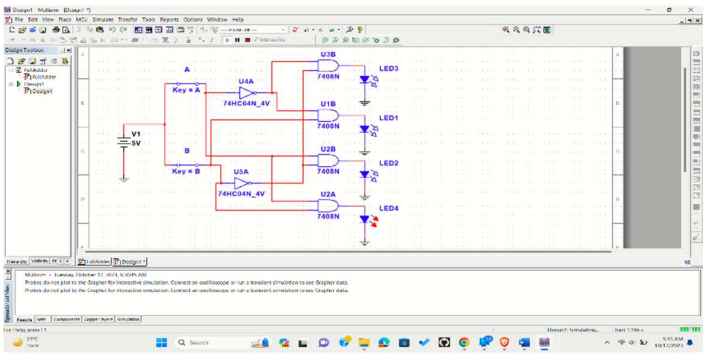
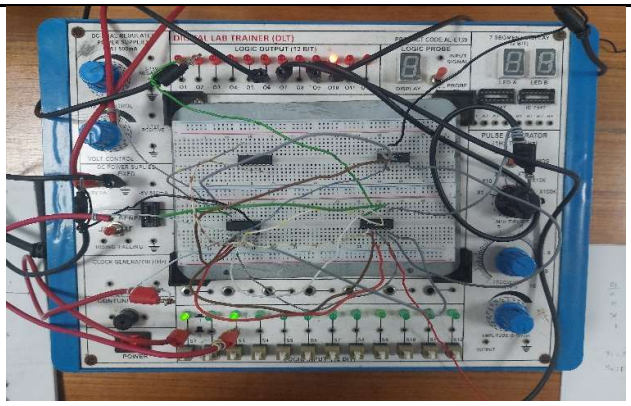


Fig.1: 2-to-4 line decoder

SIMULATION	HARDWARE
 A=0; B=0; D_0=1; $D_1 = 0; D_2 = 0; D_3 = 0$	
 A=0; B=0; D_0=0; $D_1 = 1; D_2 = 0; D_3 = 0$	
 A=0; B=0; D_0=0; $D_1 = 0; D_2 = 1; D_3 = 0$	

SIMULATION	HARDWARE
 A=0; B=0; D_0=0; D_1 = 0; D_2 = 0; D_3 =1	

Decimal to BCD encoder

Truth table for decimal to BCD encoder is given below –

Dec.	Y3	Y2	Y1	Y0
D0	0	0	0	0
D1	0	0	0	1
D2	0	0	1	0
D3	0	0	1	1
D4	0	1	0	0
D5	0	1	0	1
D6	0	1	1	0
D7	0	1	1	1
D8	1	0	0	0
D9	1	0	0	1

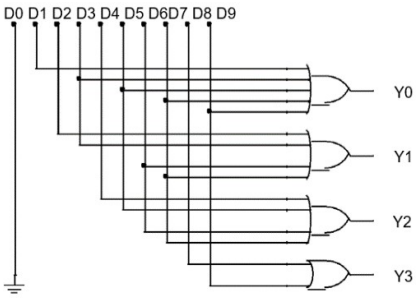
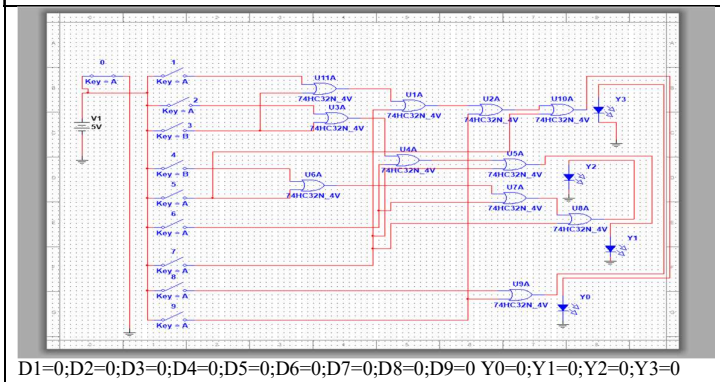
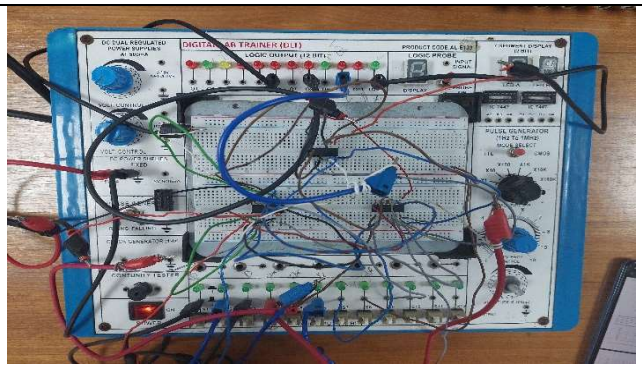
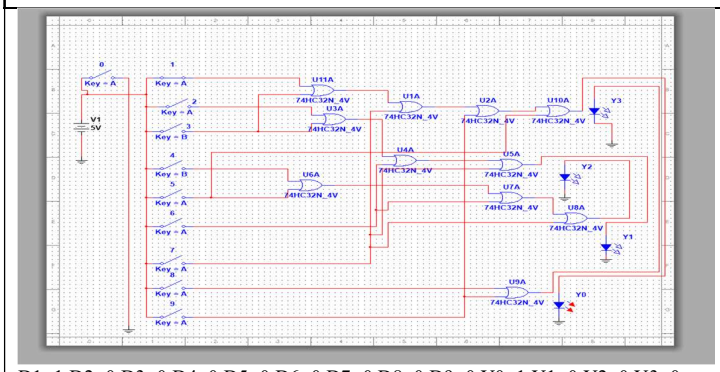
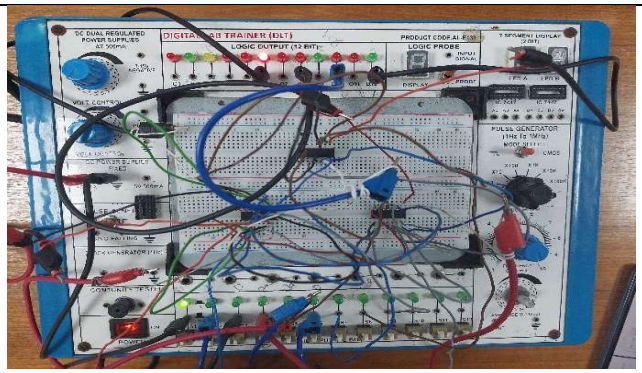
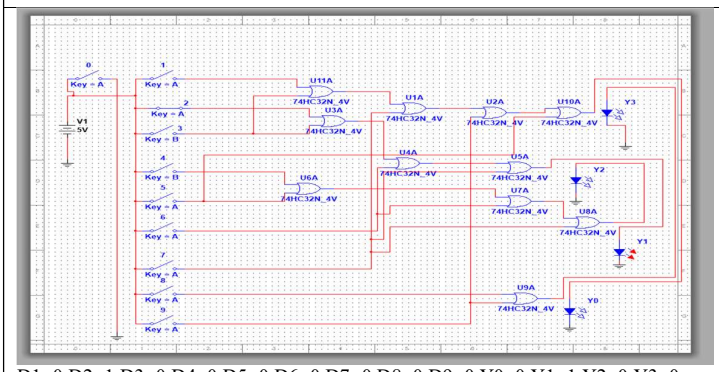
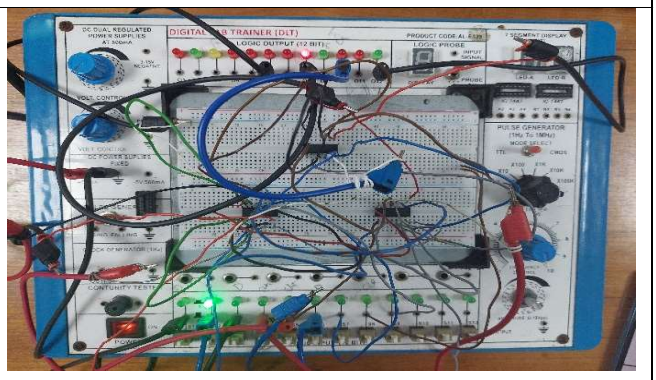
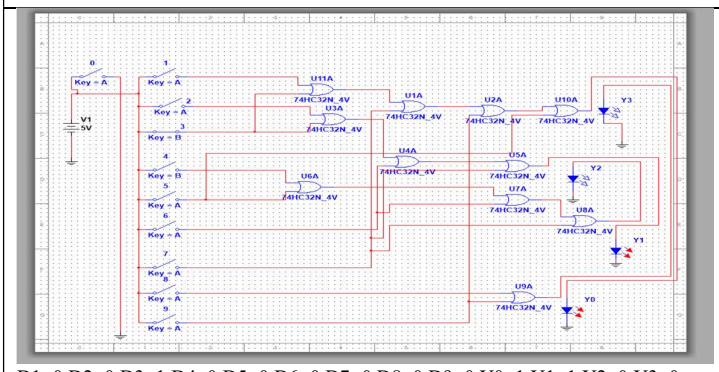
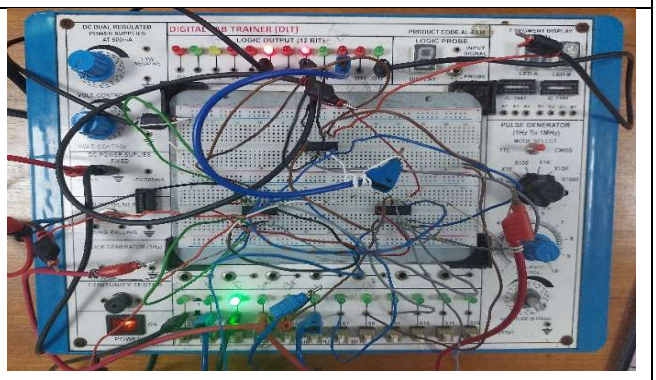
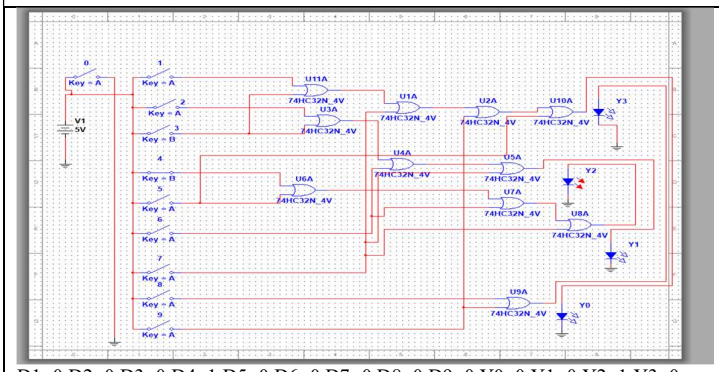
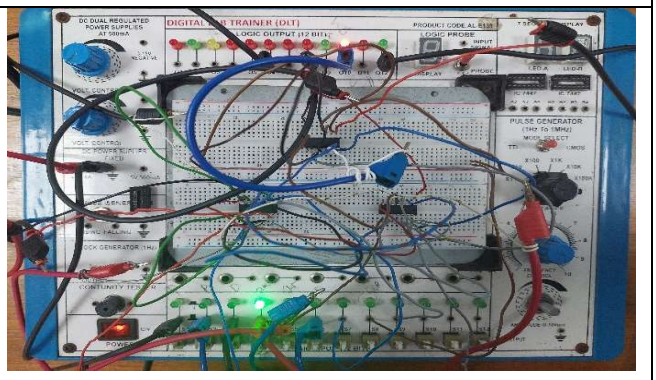
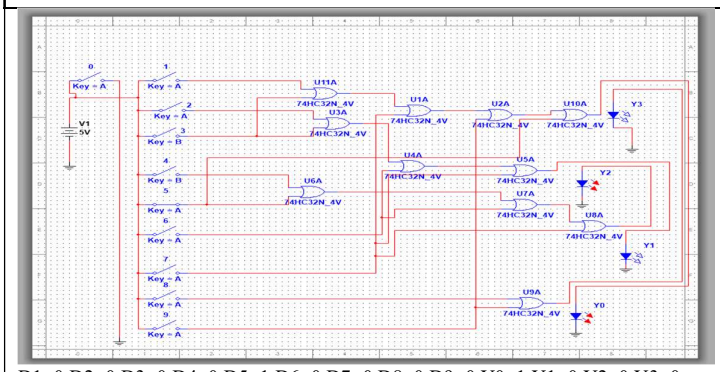
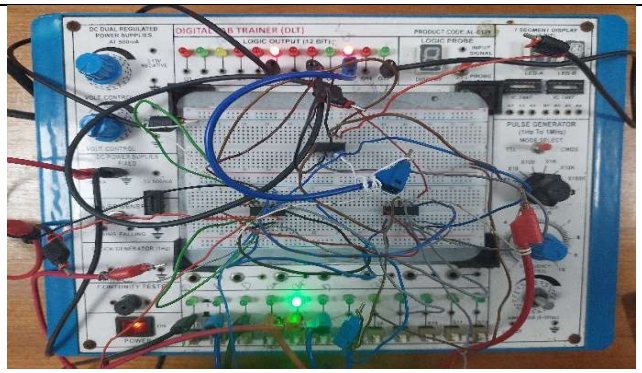
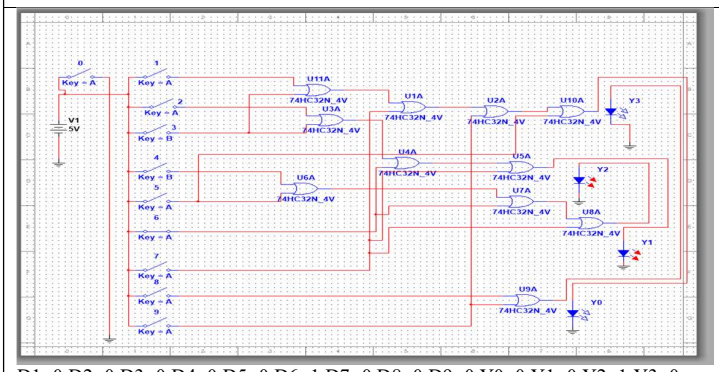
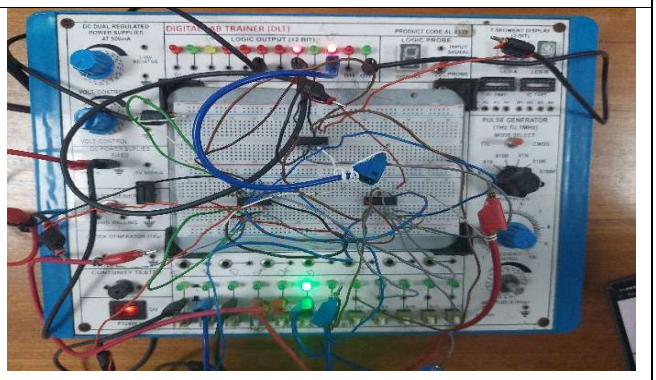
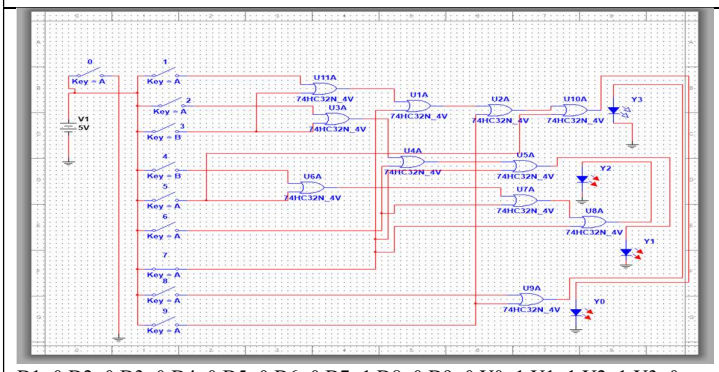
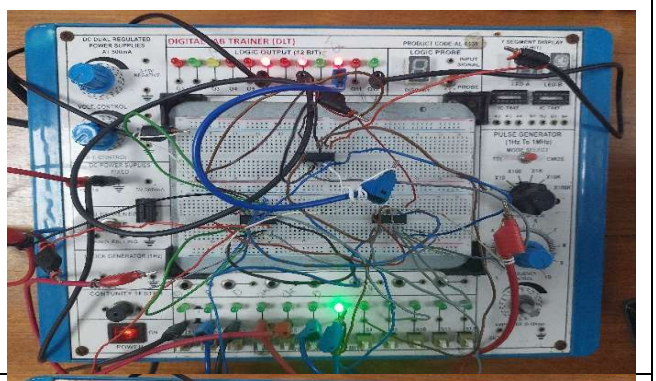
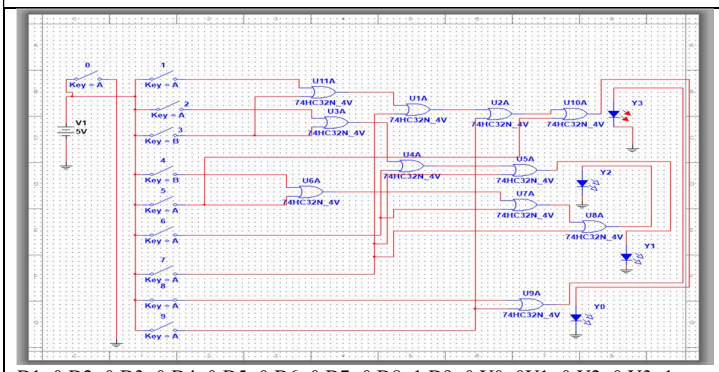
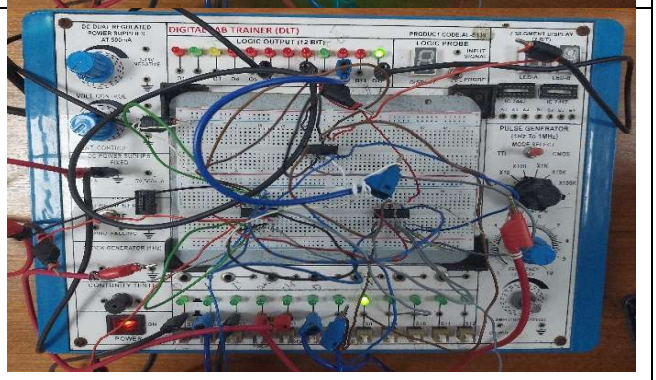
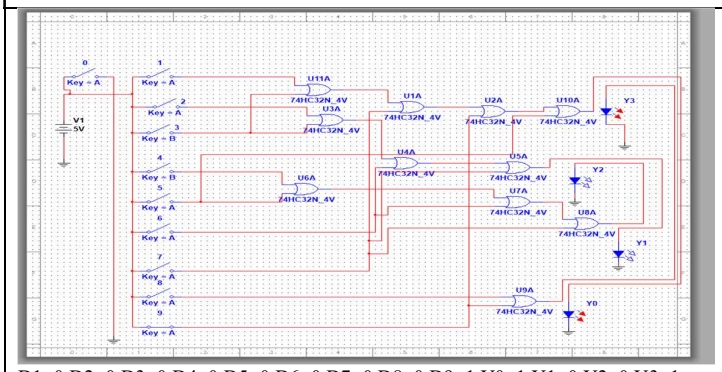
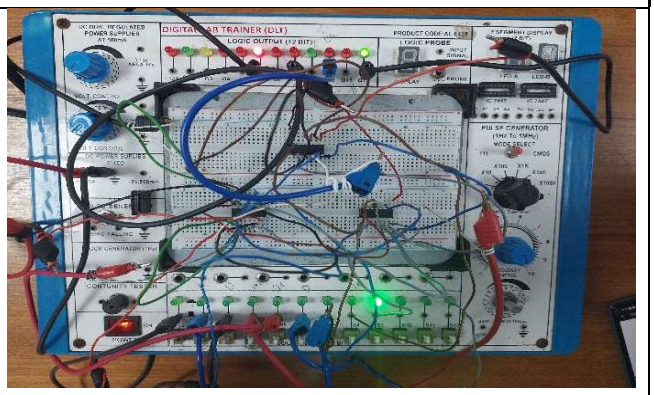


Fig.2: Decimal to BCD encoder

SIMULATION	HARDWARE
 D1=0;D2=0;D3=0;D4=0;D5=0;D6=0;D7=0;D8=0;D9=0 Y0=0;Y1=0;Y2=0;Y3=0	

SIMULATION	HARDWARE
 D1=1;D2=0;D3=0;D4=0;D5=0;D6=0;D7=0;D8=0;D9=0 Y0=1;Y1=0;Y2=0;Y3=0	
 D1=0;D2=1;D3=0;D4=0;D5=0;D6=0;D7=0;D8=0;D9=0 Y0=0;Y1=1;Y2=0;Y3=0	
 D1=0;D2=0;D3=1;D4=0;D5=0;D6=0;D7=0;D8=0;D9=0 Y0=1;Y1=1;Y2=0;Y3=0	
 D1=0;D2=0;D3=0;D4=1;D5=0;D6=0;D7=0;D8=0;D9=0 Y0=0;Y1=0;Y2=1;Y3=0	

SIMULATION	HARDWARE
 D1=0;D2=0;D3=0;D4=0;D5=1;D6=0;D7=0;D8=0;D9=0 Y0=1;Y1=0;Y2=0;Y3=0	
 D1=0;D2=0;D3=0;D4=0;D5=0;D6=1;D7=0;D8=0;D9=0 Y0=0;Y1=0;Y2=1;Y3=0	
 D1=0;D2=0;D3=0;D4=0;D5=0;D6=0;D7=1;D8=0;D9=0 Y0=1;Y1=1;Y2=1;Y3=0	
 D1=0;D2=0;D3=0;D4=0;D5=0;D6=0;D7=0;D8=1;D9=0 Y0=0Y1=0;Y2=0;Y3=1	

SIMULATION	HARDWARE
 D1=0;D2=0;D3=0;D4=0;D5=0;D6=0;D7=0;D8=0;D9=1 Y0=1;Y1=0;Y2=0;Y3=1	

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

Truth table for this priority encoder is given below.

A3	A2	A1	A0	Y1	Y0
x	1	x	x	1	0
x	0	1	x	0	1
1	0	0	x	1	1
0	0	0	1	0	0

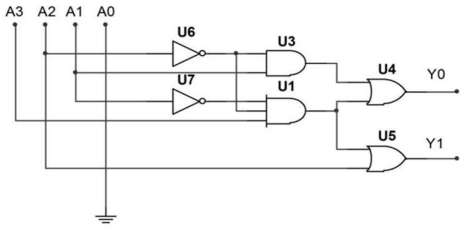
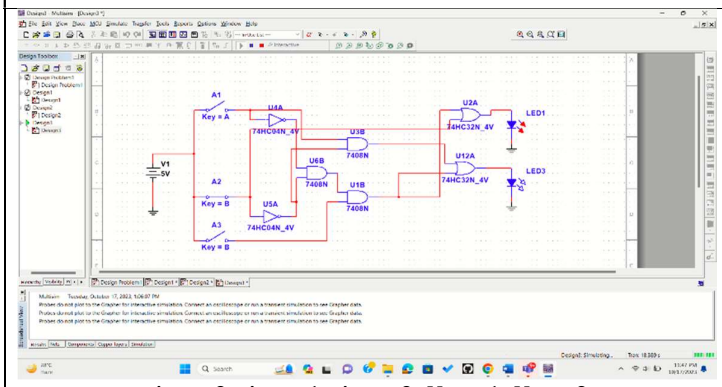
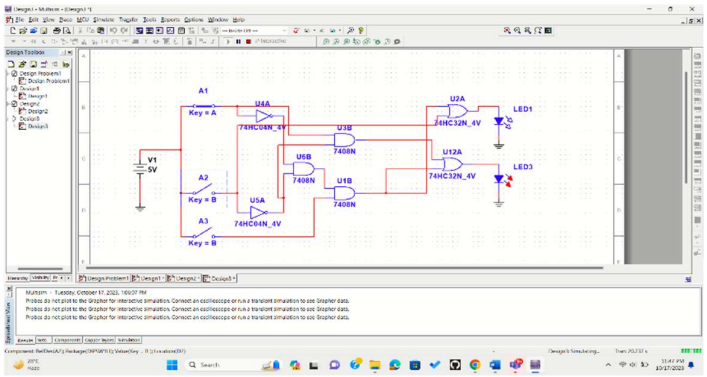
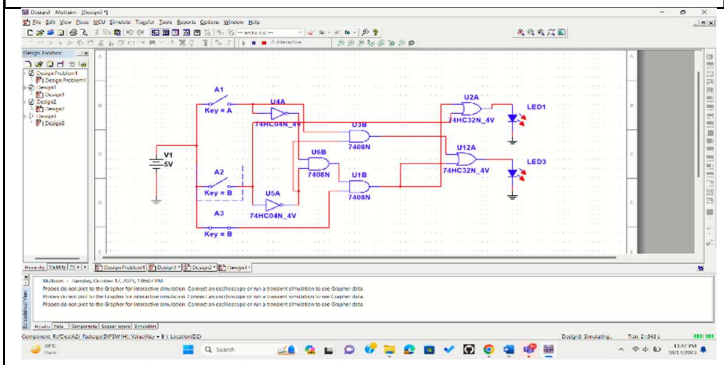
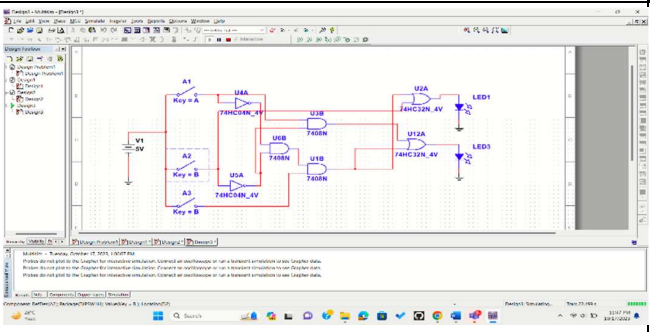


Fig .3: 4-to 2 priority encoder with a priority sequence of 2,1,3,0

SIMULATION	SIMULATION
 $A_3 = 0; A_2 = 1; A_1 = 0; Y_1 = 1; Y_2 = 0$	 $A_3 = 0; A_2 = 1; A_1 = 0; Y_1 = 1; Y_2 = 0$

SIMULATION	HARDWARE
	
$A_3 = 0; A_2 = 1; A_1 = 0; Y_1 = 1; Y_2 = 0$	$A_3 = 0; A_2 = 1; A_1 = 0; Y_1 = 1; Y_2 = 0$

Discussion

The main purpose of this experiment is to understand the functionality of MUX, DEMUX, Encode and decoder circuits. 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. 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. So throughout the experiment, we simulated the circuits of these circuits in multisim software and verified it with the truth table. Then we implemented the circuit in trainer board. While implementing the hardware oart, there were some difficulties we faced like loose connection, problem in the output led, and problem in the input switches. We investigated these issues and solved them.

Reference:

http://www.tutorialspoint.com/computer_logical_organization/combinational_circuits.htm

Part I

4to1 Multiplexer

Table:1

S1	S0	f
0	0	D0
0	1	D1
1	0	D2
1	1	D3

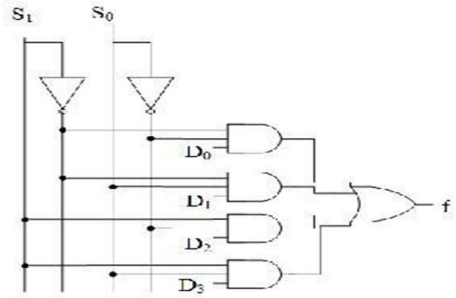
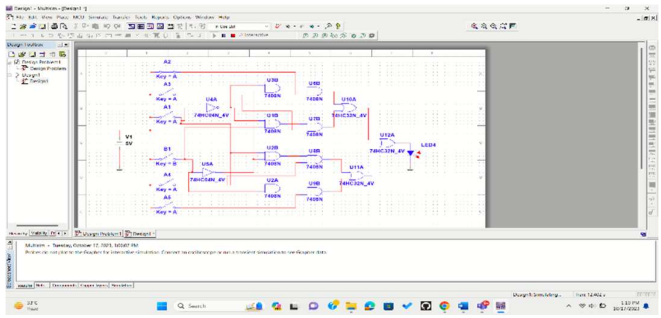
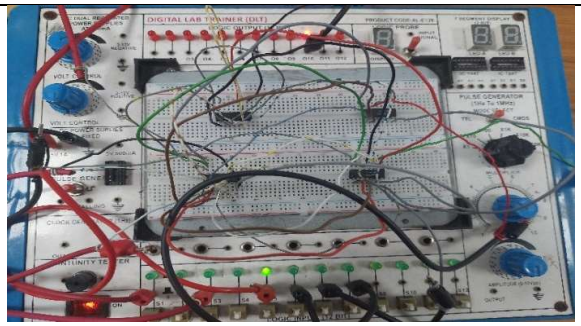
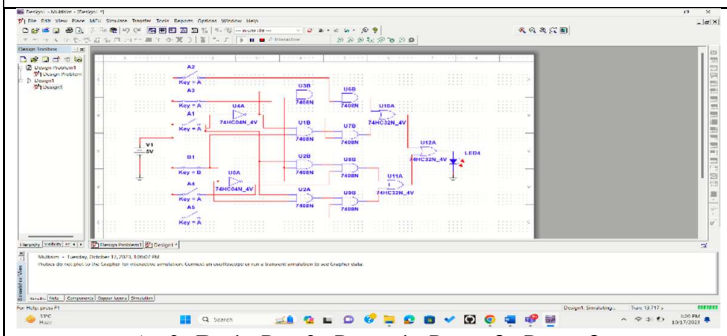
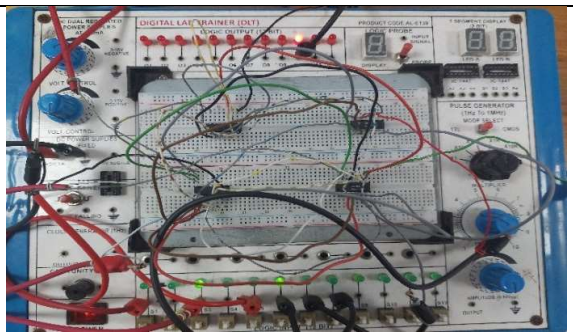
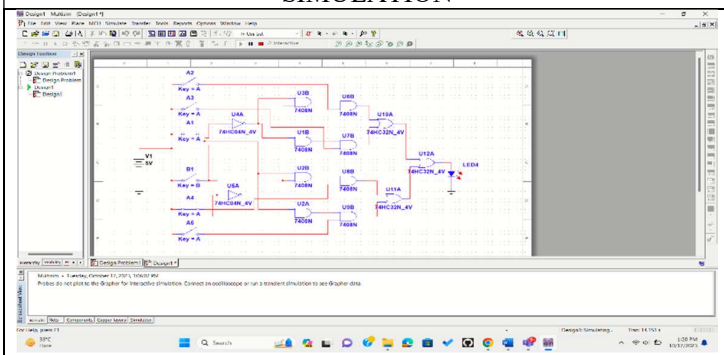
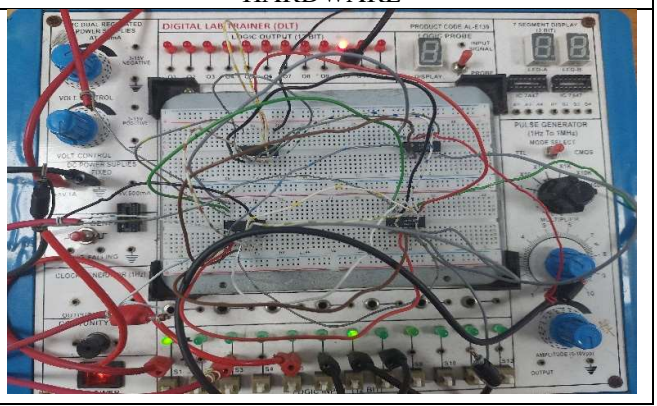
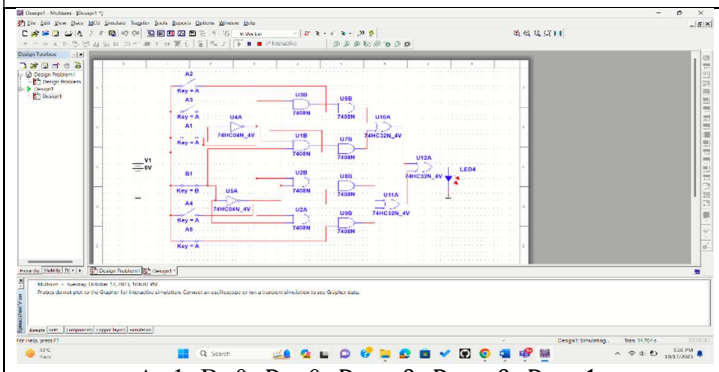
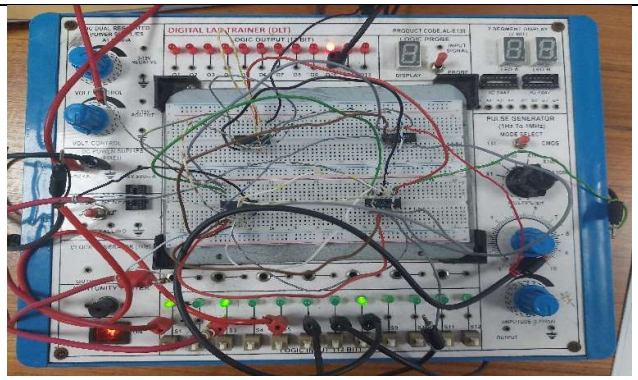


Figure1: 4to1 Multiplexer

SIMULATION	HARDWARE
 A=0; B=0; D_0=1; $D_1 = 0; D_2 = 0; D_3 = 0$	
 A=0; B=1; D_0=0; $D_1 = 1; D_2 = 0; D_3 = 0$	
SIMULATION	HARDWARE
 A=1; B=0; D_0=1; $D_1 = 0; D_2 = 1; D_3 = 0$	

SIMULATION	HARDWARE
 A=1; B=0; D_0=0; $D_1 = 0; D_2 = 0; D_3$ =1	

Demultiplexer

1 to 4 Demultiplexer

Table:2

S1	S0	D0	D1	D2	D3
0	0	Din	0	0	0
0	1	0	Din	0	0
1	0	0	0	Din	0
1	1	0	0	0	Din

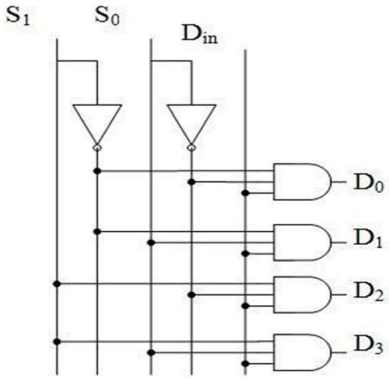
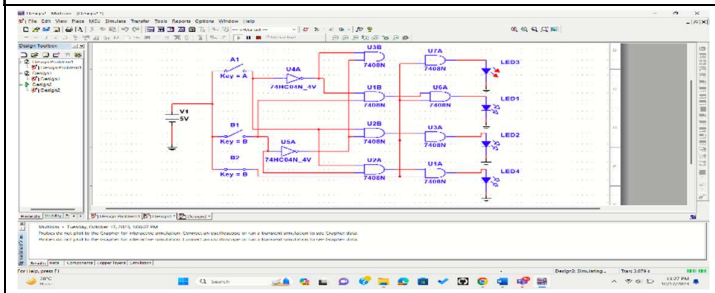
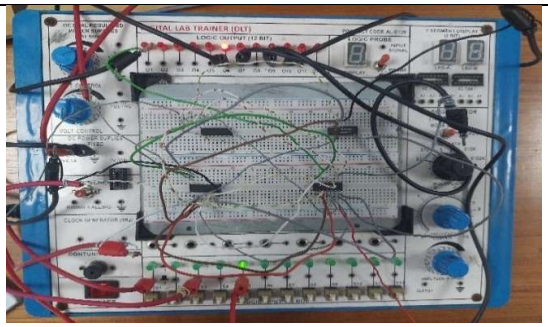
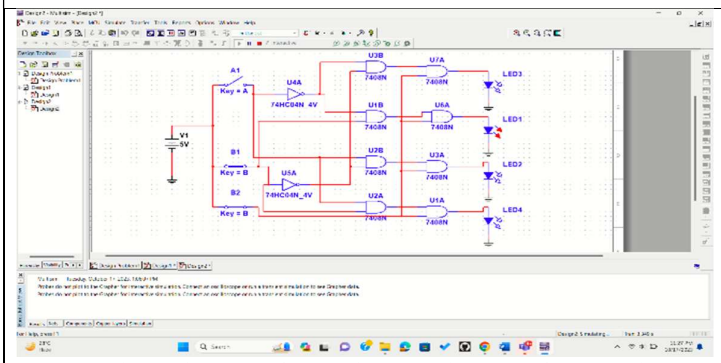
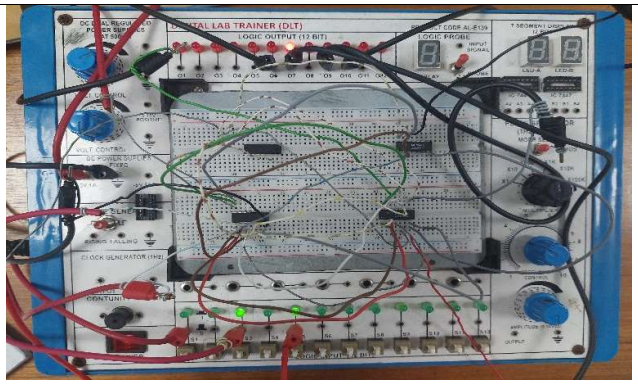
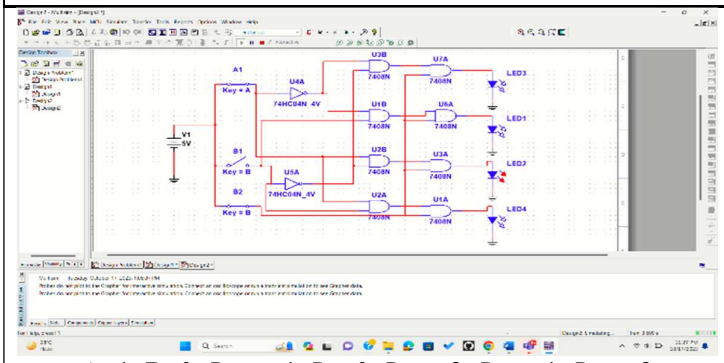
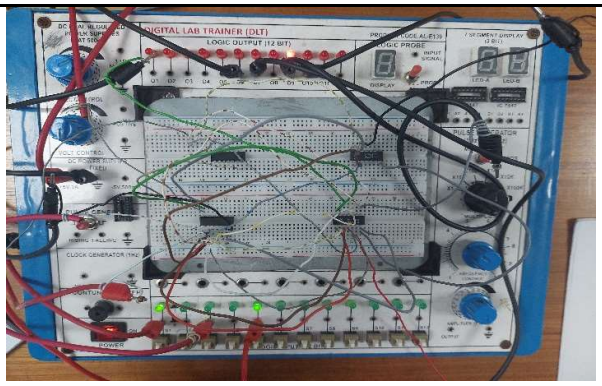
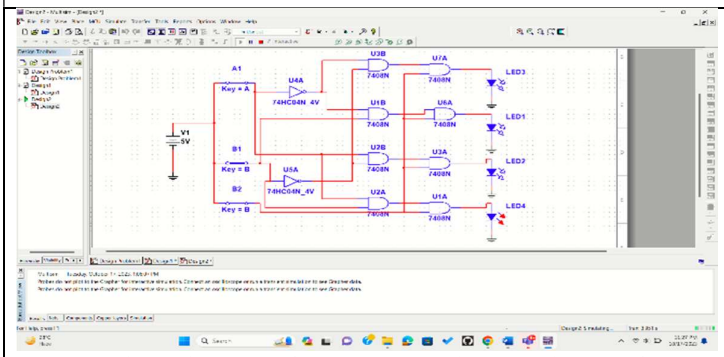


Figure 2: 1 to 4 Demultiplexer

SIMULATION	HARDWARE
 A=0; B=0; $D_{in} = 1; D_0$=1; $D_1 = 0; D_2 = 0; D_3 = 0$	

SIMULATION	HARDWARE
 A=0; B=1;$D_{in} = 1$; D_0=1; $D_1 = 0$; $D_2 = 1$; $D_3 = 0$	
 A=1; B=0; $D_{in} = 1$; D_0=0; $D_1 = 0$; $D_2 = 1$; $D_3 = 0$	
 A=1; B=1; $D_{in} = 1$; D_0=0; $D_1 = 0$; $D_2 = 0$; D_3 =1