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

Abstract

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.

Part I
Introduction (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.

Theory and Methodology

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 S1 and S0. Here, the number of selection pinis 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, ifS_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^n to1 line MUX. The MUX we are going to design is a 4 to 1 MUX. There could be also 2 to 1 MUX, 8 to 1 MUX, 16 to 1 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

S1	S0	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-

$f = S1. \bar{S}_0 . \bar{D}_0 + S_1. \bar{S}_0 . D_1 + S_1. S_0 . \bar{D}_2 + S_1. S_0. D_3$ ....(1)

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

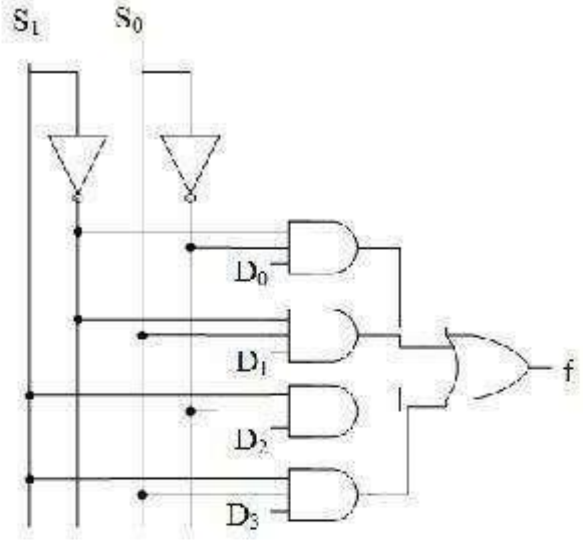


Figure 1: 4 to 1 Multiplexer

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 1 to 2, 1 to 4 and 1 to 8 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 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}	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-

$D_0 = \bar{S}_1 \,.\, \bar{S}_0 \,.\, D_{in}$........................(2)

$D_1 = \bar{S}_1 \,.\, S_0 \,.\, D_{in}$..................................(3)

$D_2 = S_1 .\, \bar{S}_0 \,.\, D_{in}$......................................(4)

$D_3 = S_1 .\, \bar{S}_0 \,.\, D_{in}$......................................(5)

The circuit for 1to4 line DEMUX is given below-

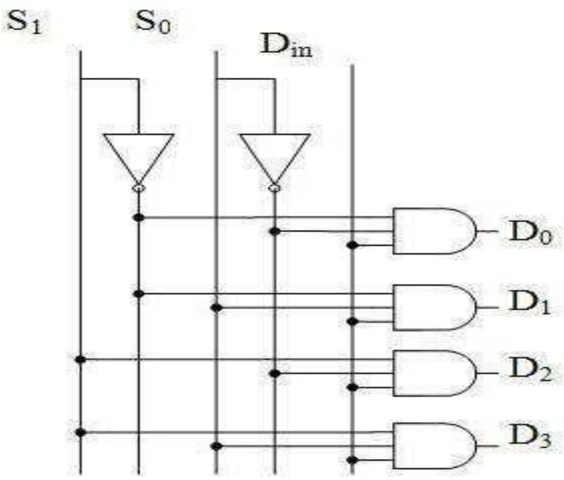


Figure 2: 1 to 4 Demultiplexer

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

Part II: Encoder and Decoder:

An encoder is a device or a circuit that converts information from one format or code to another. A decoderdoes 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, softwareprograms, 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.

Decoder:

A decoder can convert binary information from n input lines to a maximum of 2^n unique output lines. The2-to-4 line decoder will take inputs from two lines and convert them to 4 lines.

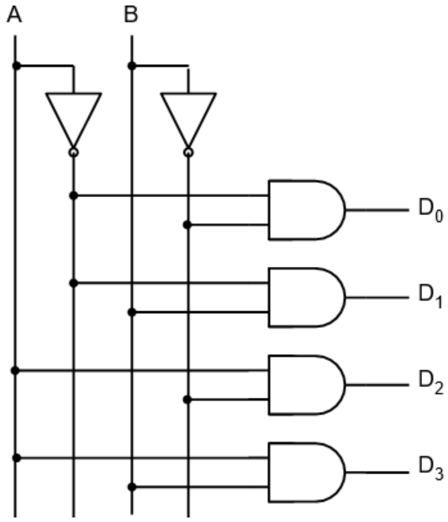


Fig. 5: 2-to-4 line decoder

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

$$D_0 = \bar{A}\bar{B} \qquad (6)$$

$$D_1 = \bar{A}B \qquad (7)$$

$$D_2 = A\bar{B} \qquad (8)$$

$$D_3 = AB \qquad (9)$$

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

Priority encoder:

A priority encoder is a circuit or algorithm that compresses multiple binary inputs into a smaller numberof 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 highestpriority 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 areignored (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.

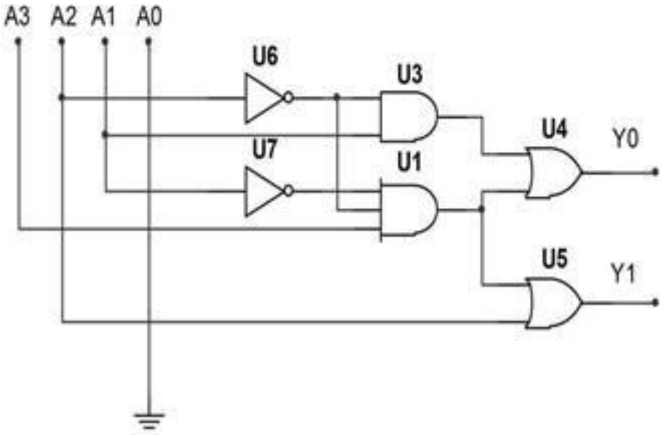


Fig. 7: 4-to 2 priority encoder with a priority sequence of 2,1,3,0

The expressions for implementing the above priority encoder–

$$Y0 = A2'.A1 + A3.A2'.A1' \quad Y1 = A2 + A3.A2'.A1'$$

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:

SL	Apparatus	Picture	Quantity	Remarks
1.	Digital trainer board.		1	Good condition
2.	Integrated Circuits (ICs) 7400 7402 7404 7408 7432 7486		 1 1 1 1 1 1	Was not in good condition. Had to change it.
3.	Power Supply		1	Good condition
4.	Connecting wires		multiple	Good condition

Experimental Procedure:

1. Connected the circuit according to the figures.
2. Used the toggle switches on the trainer board to provide input signal to the circuits. Connected the outputs to the LEDs on the multisim.
3. Applied the input signals and observed and noted the corresponding output signals.

Hardware Setup :

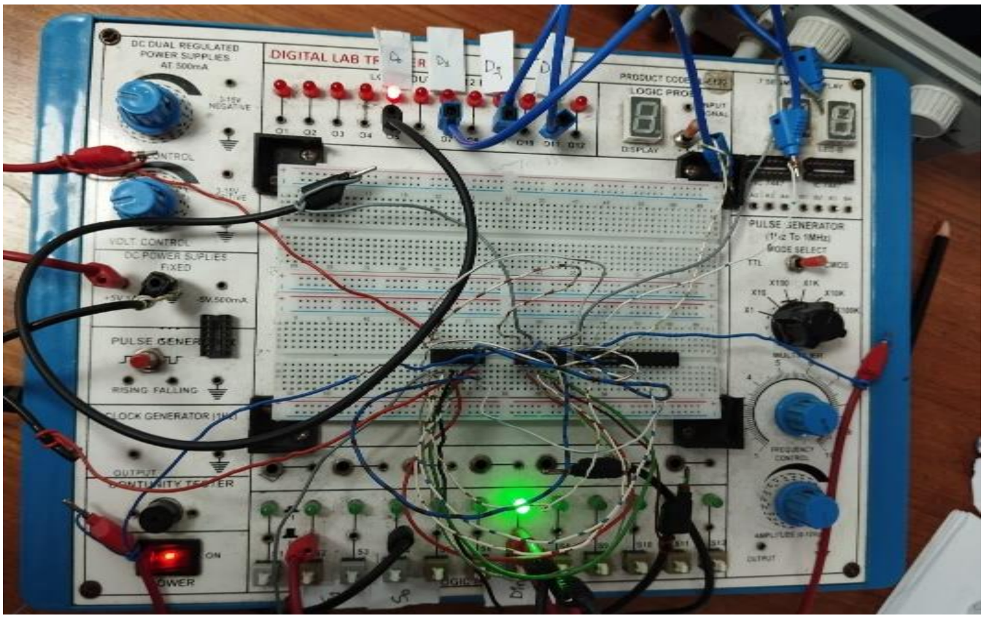


Fig-01 : 4 to 1 Multiplexer

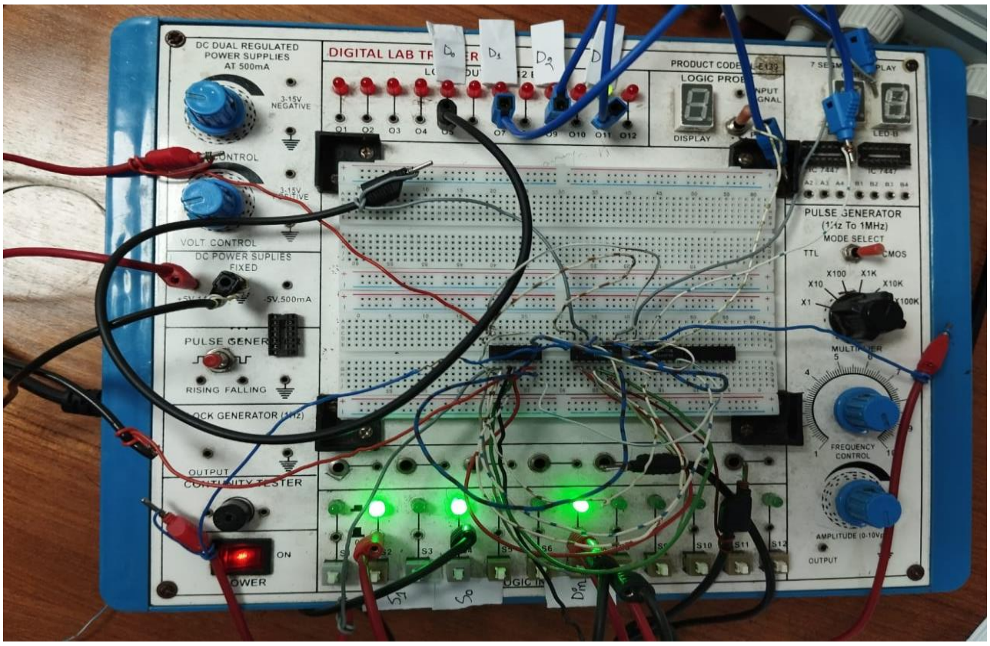


Fig-02 : 1 to 4 De-Multiplexer

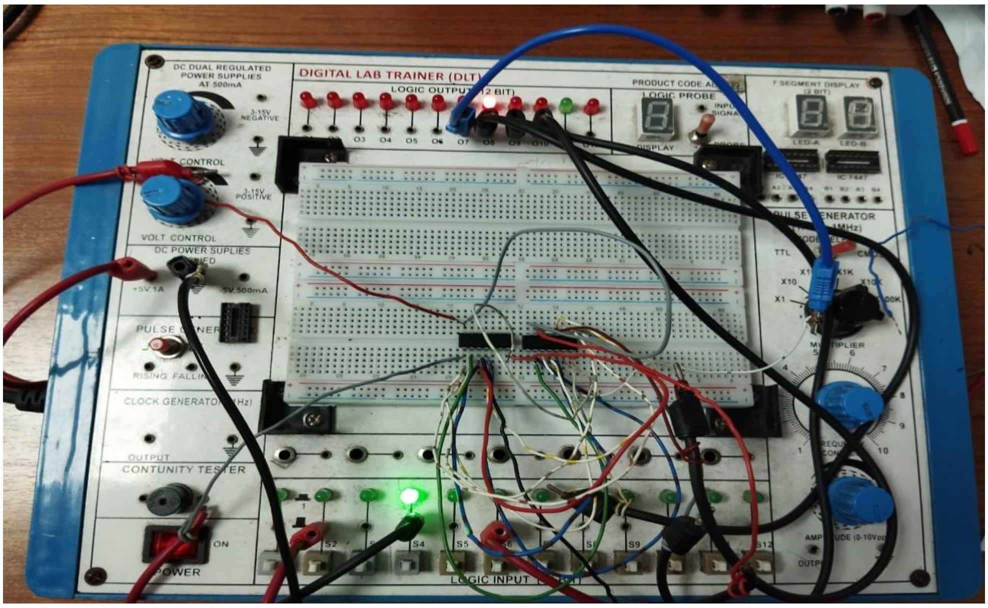


Fig-03 : 2 to 4 Decoder

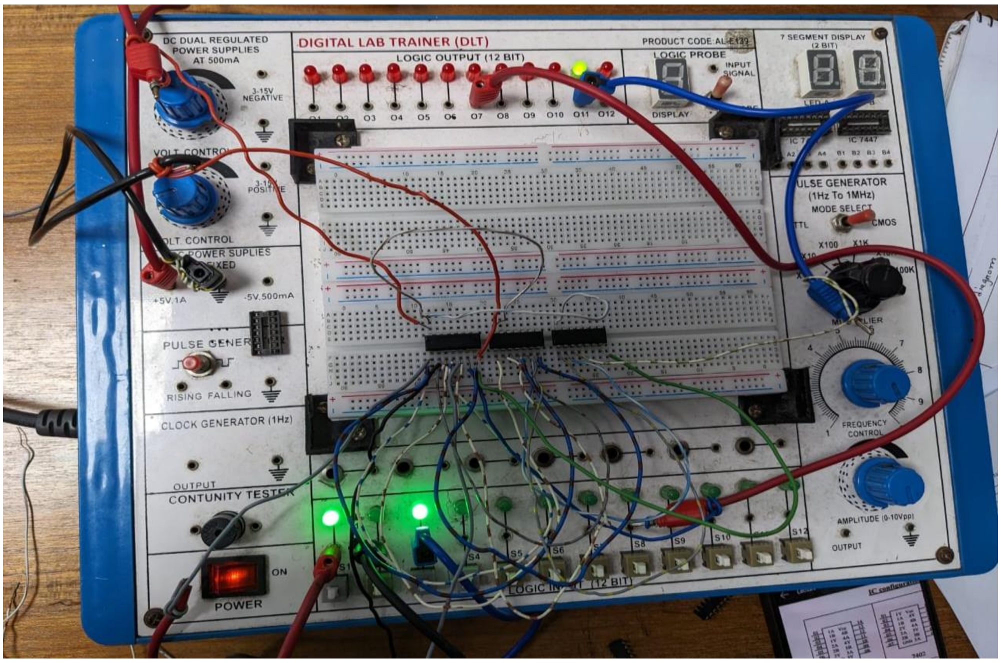
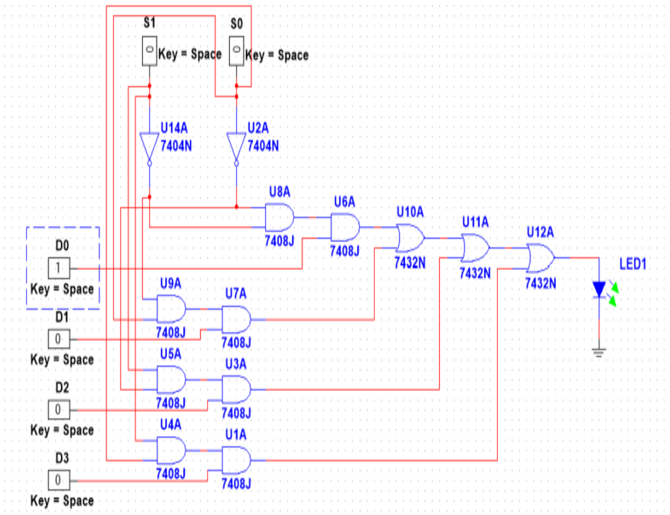


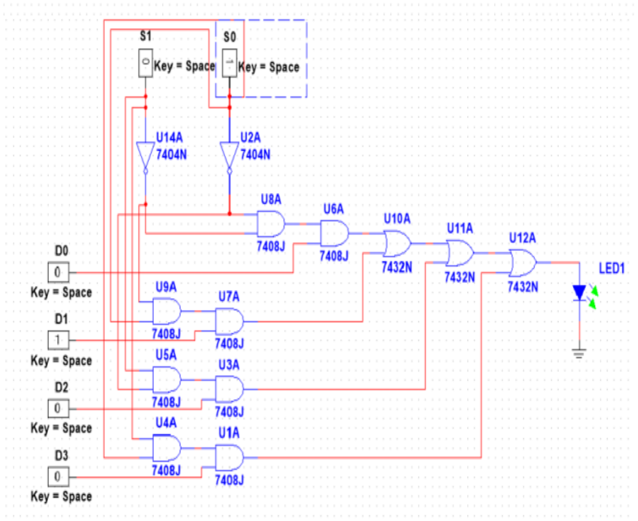
Fig-04 : 4 to 2 Priority Encoder

Simulation:

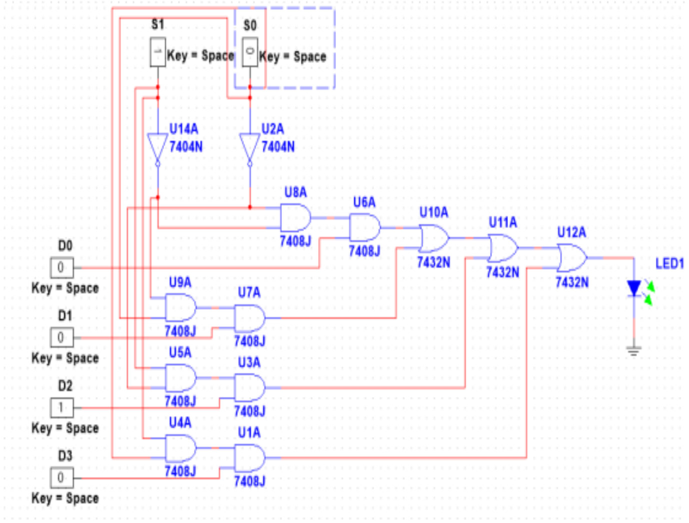
1) Simulation of 4 to 1 Multiplexer :



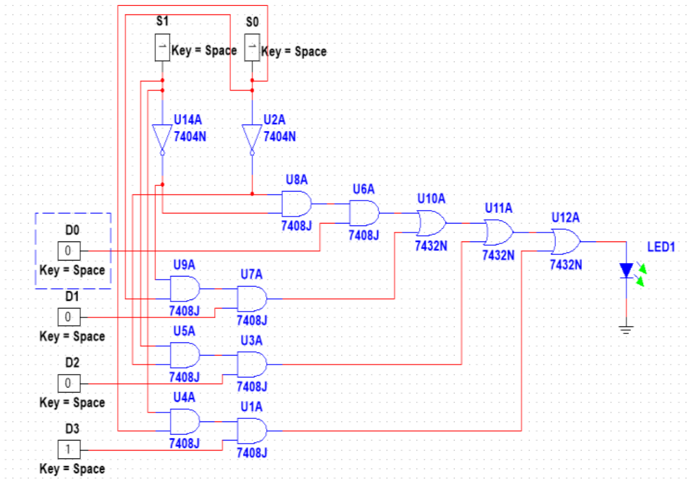
a) Fig-05 : D0=1, D1=D2=D3=0



b) Fig-06 : D0=D2=D3=0, D1=1

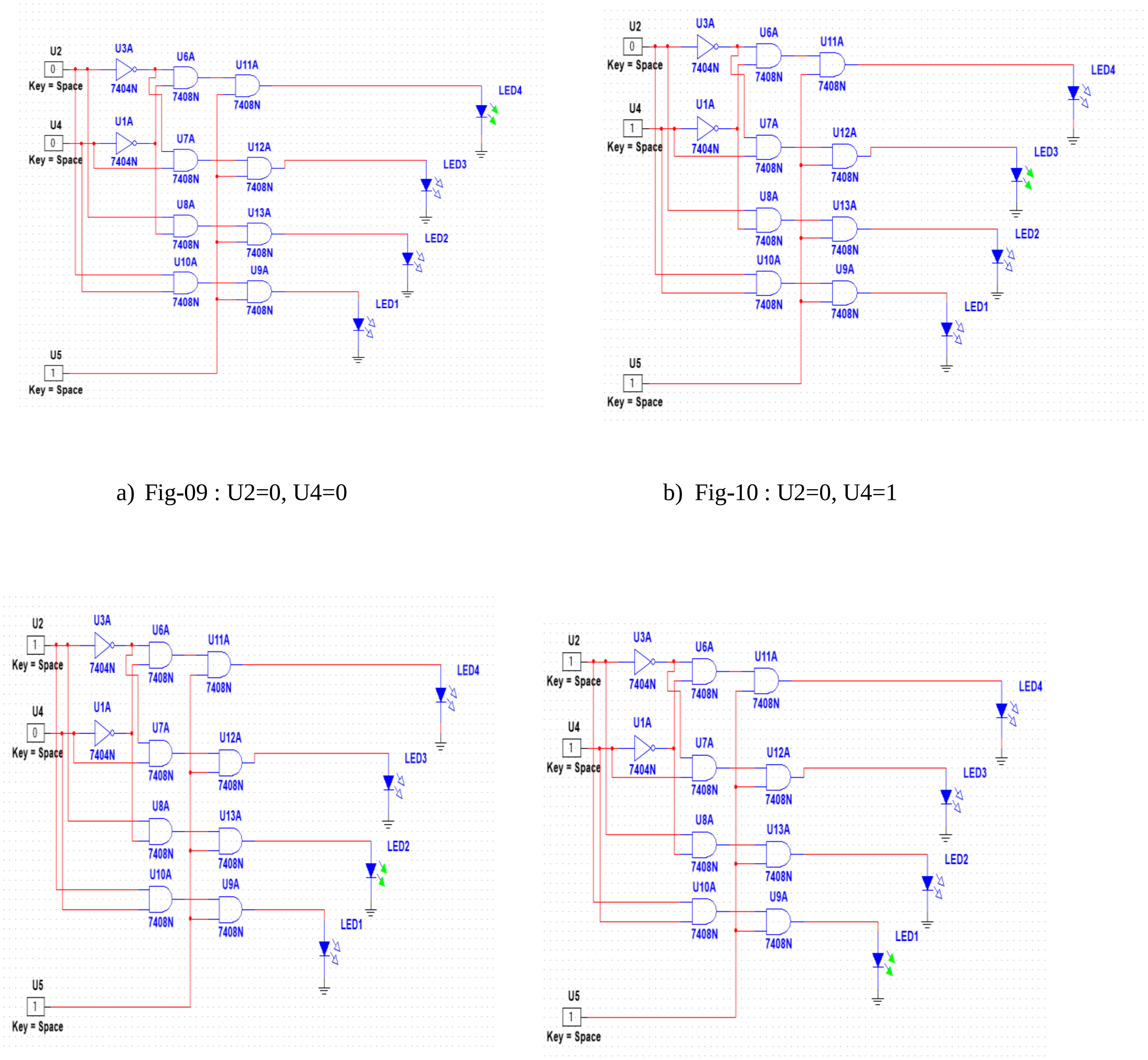


c) Fig-07 : D0=D1=D3=0, D2=1



d) Fig-08 : D0=D1=D2=0, D3=1

2)Figure: Simulation of 1 TO 4 Demultiplexer



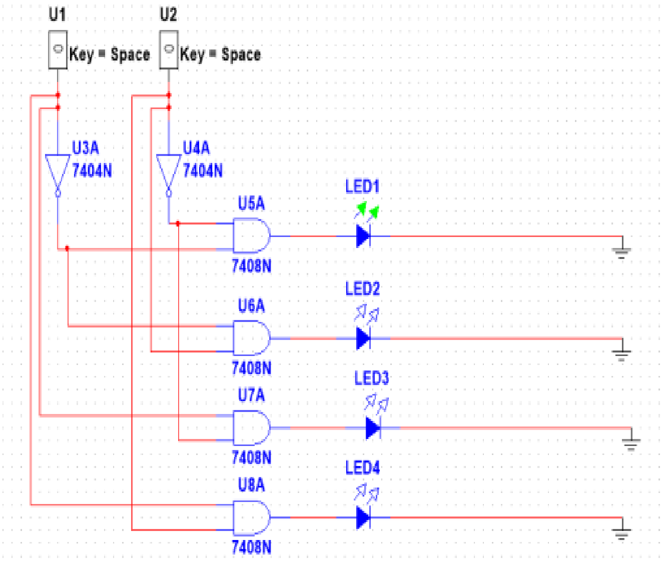
a) Fig-09 : U2=0, U4=0

b) Fig-10 : U2=0, U4=1

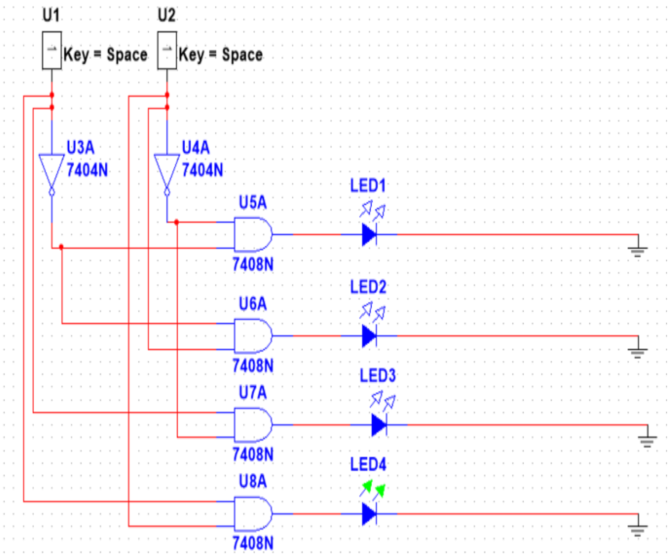
c) Fig-11 : U2=1, U4=0

d) Fig-12 : U2=1, U4=1

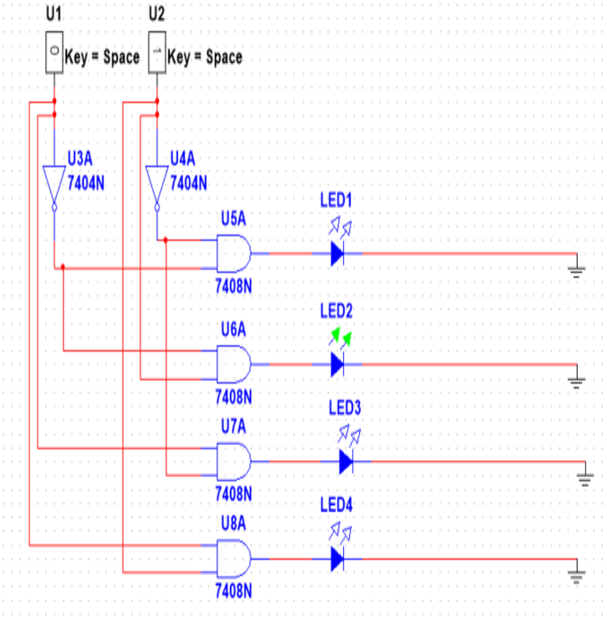
3)Figure: Simulation of 2 to 4 Decoder :



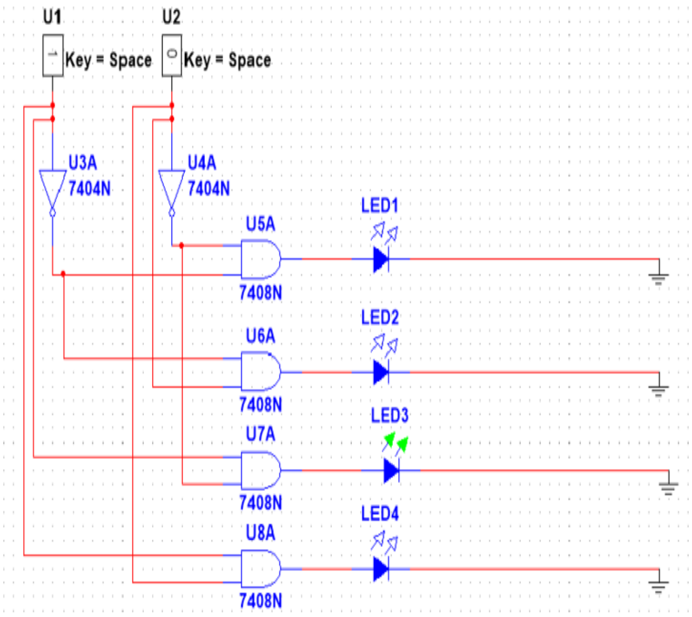
a) Fig-13 : U1=0, U2=0



b) Fig-14 : U1=1, U2=1

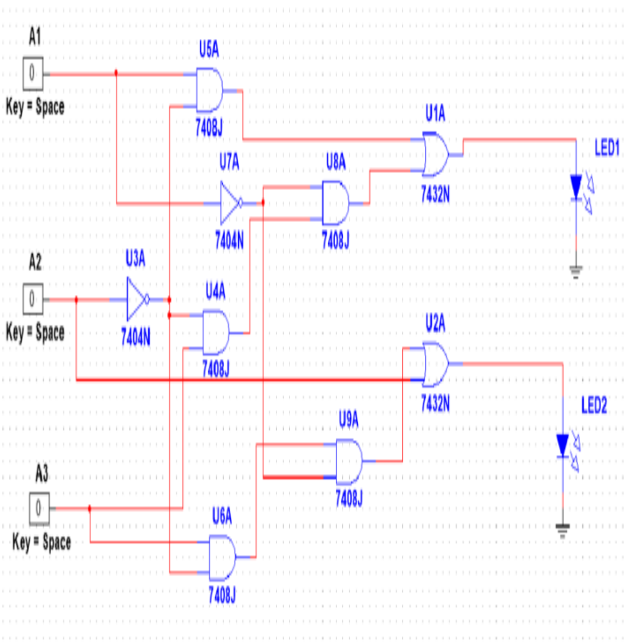


c) Fig-15 : U1=0, U2=1

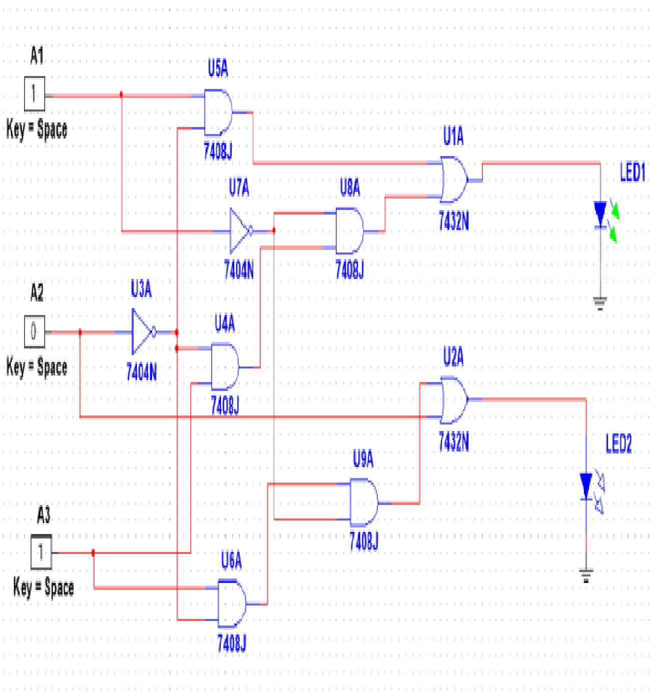


d) Fig-16 : U1=1, U2=0

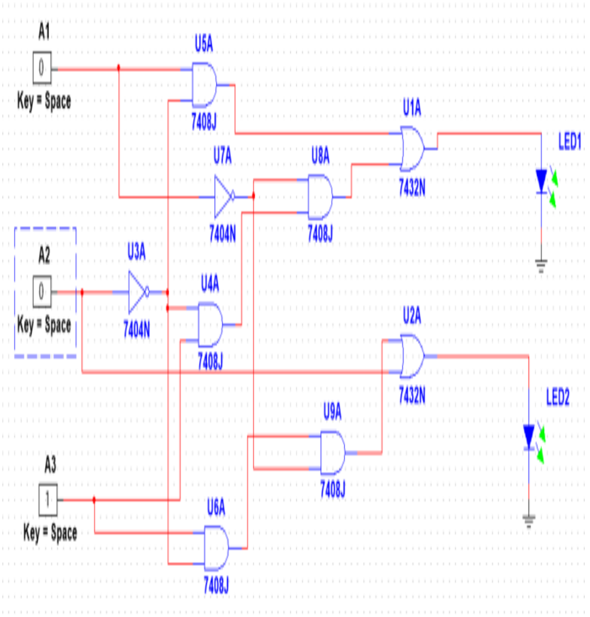
4) Simulation of 4 to 2 Priority Encoder :



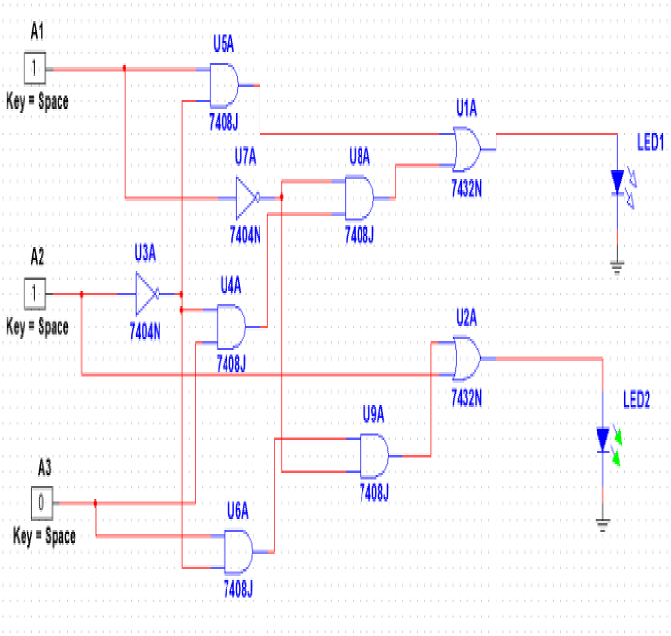
a) Fig-17 : A1=0, A2=0, A3=0



b) Fig-18 : A1=1, A2=0, A3=1



c) Fig-19 : A1=0, A2=0, A3=1



d) Fig-20 : A1=1, A2=1, A3=0

Discussion

The data and findings indicate that the truth tables derived from the executed MUX, DEMUX, and Decoder circuits aligned perfectly with the theoretical results for every possible input data combination. No discrepancies or errors were detected in the circuit implementations or simulations, as thorough checks were conducted on all circuits.

Conclusion

In accordance with the established objectives, the MUX, DEMUX, and Decoder circuits were examined, deduced, and constructed. The operational principles of these circuits were closely observed, and the realized implementations were validated against the theoretical expectations. Consequently, the experiment can be deemed successful.

Reference(s):

[1] www.tutorialspoint.com

[2] www.electronics-tutorials.ws

[3] faculty.kfupm.edu.sa

[4] "Digital Fundamentals" by Thomas L. Floyd

[5] American International University–Bangladesh (AIUB) Digital Logic And Circuits Lab Manual.