AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

DIGITAL LOGIC AND CIRCUITS LAB

Summer 2022-2023

Section: F
Group Number: 02

EXPERIMENT NO. 4

NAME OF THE EXPERIMENT

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

Supervised By

SHAHRIYAR MASUD RIZVI

Faculty of Engineering, AIUB

Submitted By:

Name of the Student	ID Number
1. NOKIBUL ARFIN SIAM	21-44793-1
2. AHNAF AHMED	20-42173-1
3. SAIFUL ISLAM	20-42585-1
4. MAHADI HASAN	19-39711-1
5. SHEIKH FAHIM FUAD	21-44721-1

Introduction:

In this experiment, we will learn how to design and implement multiplexers (MUX) and demultiplexers (DeMUX) of different sizes using basic logic gates. We will also learn how to construct bigger multiplexer using smaller multiplexers and 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 D0 ,D1, D2 and D3. 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 D0 or D1 or D2 or D3, dependent on the values of two selection pins S1 and S0. Here, the number of selection pin is two. Four combinations are possible using these two selection pins S1 and S0, such as (S1, S0) = (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 S0 = 0 then f = D0, if S1 = 0 and S0 = 1 then f = D1, if S1 = 1 and S0 = 0 then f = D2 and if S1 = 1 and S0 = 1 then f = D3.

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 as 2nto1 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, 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.

$$f = S_1\overline{S_0}D_0 + S_1\overline{S_0}D_1 + S_1S_0\overline{D_2} + S_1S_0D_3 \ \ldots \ (1)$$

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

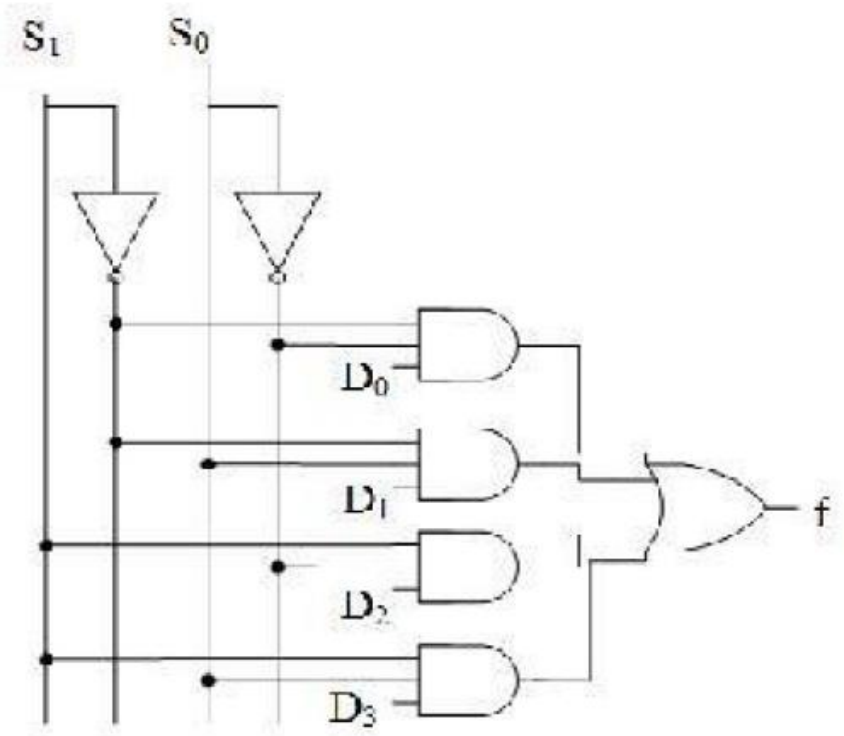


Figure1: 4to1 Multiplexer

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 Din, two selection pins S1 and S0 and four outputs D0 ,D1, D2 and D3. Now if S1 = 0 and S0 = 0 then D0 = Din, if S1 = 0 and S0 = 1 then D1 =Din, if S1 = 1 and S0 = 0 then D2 = Din and if S1 = 1 and S0 = 1 then D3 = Din. 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 D0 ,D1, D2 and D3¯… (3) as given below.

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

The circuit for 1to4 line demux is given below.

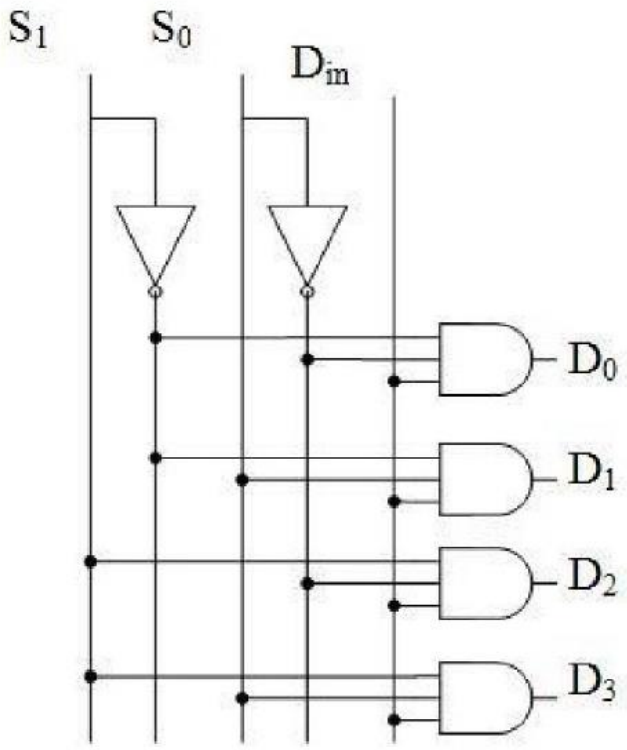


Figure 2: 1 to 4 Demultiplexer

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.

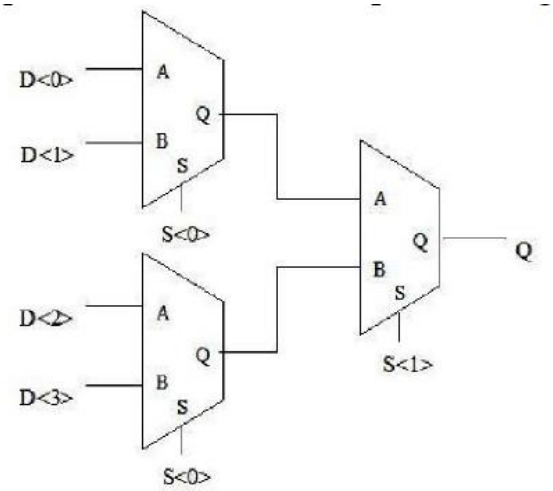


Figure 3: 4to1 multiplexer using 2to1 multiplexers.

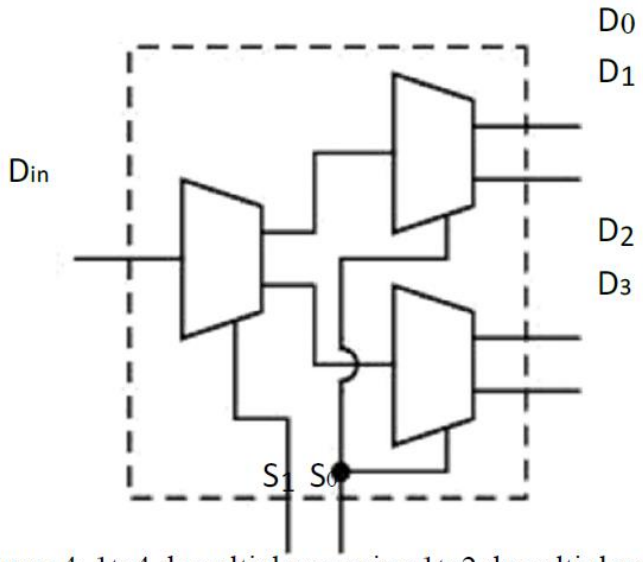


Figure 4: 1to4 demultiplexer using 1to2 demultiplexers.

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.

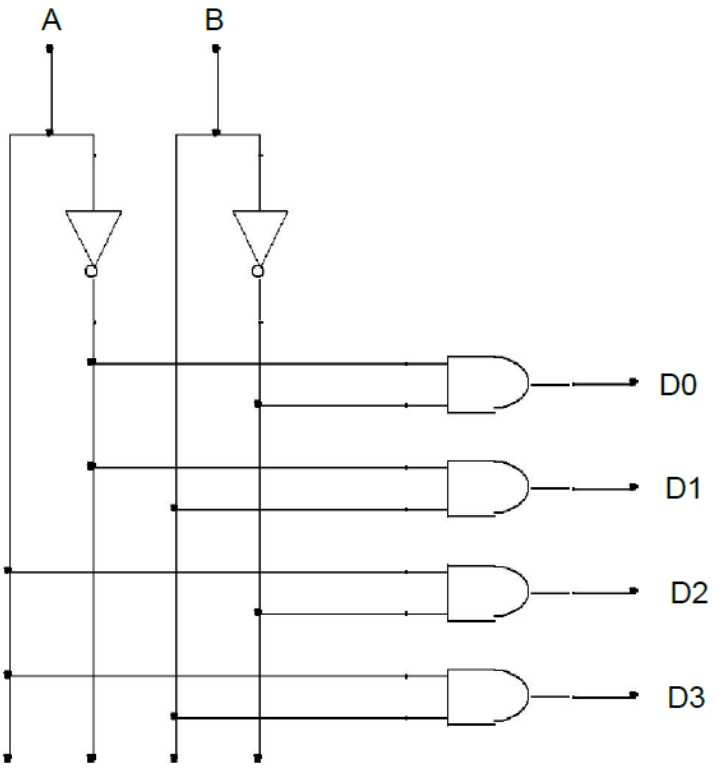


Fig.1: 2-to-4 line decoder

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).

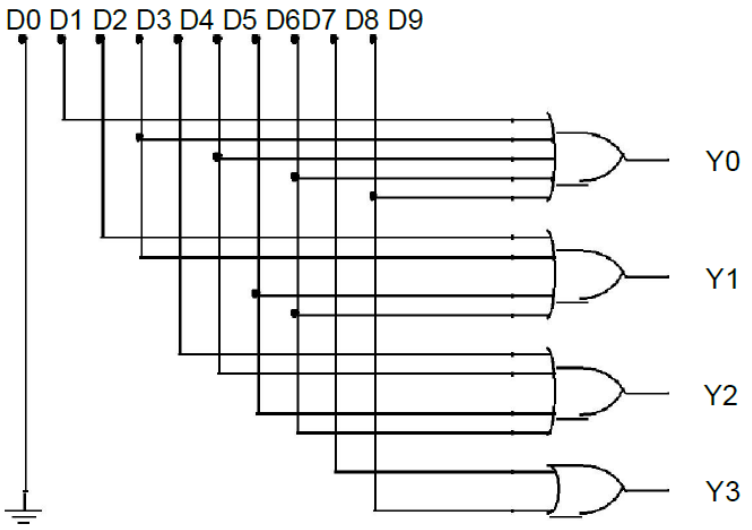


Fig.2: Decimal to BCD encoder

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., 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.

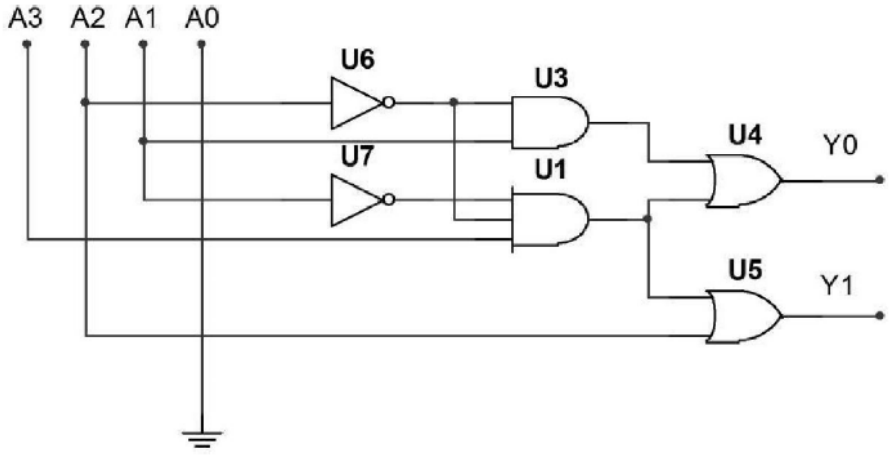


Fig .3: 4-to 2 priority encoder with a priority sequence of 2,1,3,0

The expressions for implementing the above priority encoder–

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

Truth table for this priority encoder is given below –

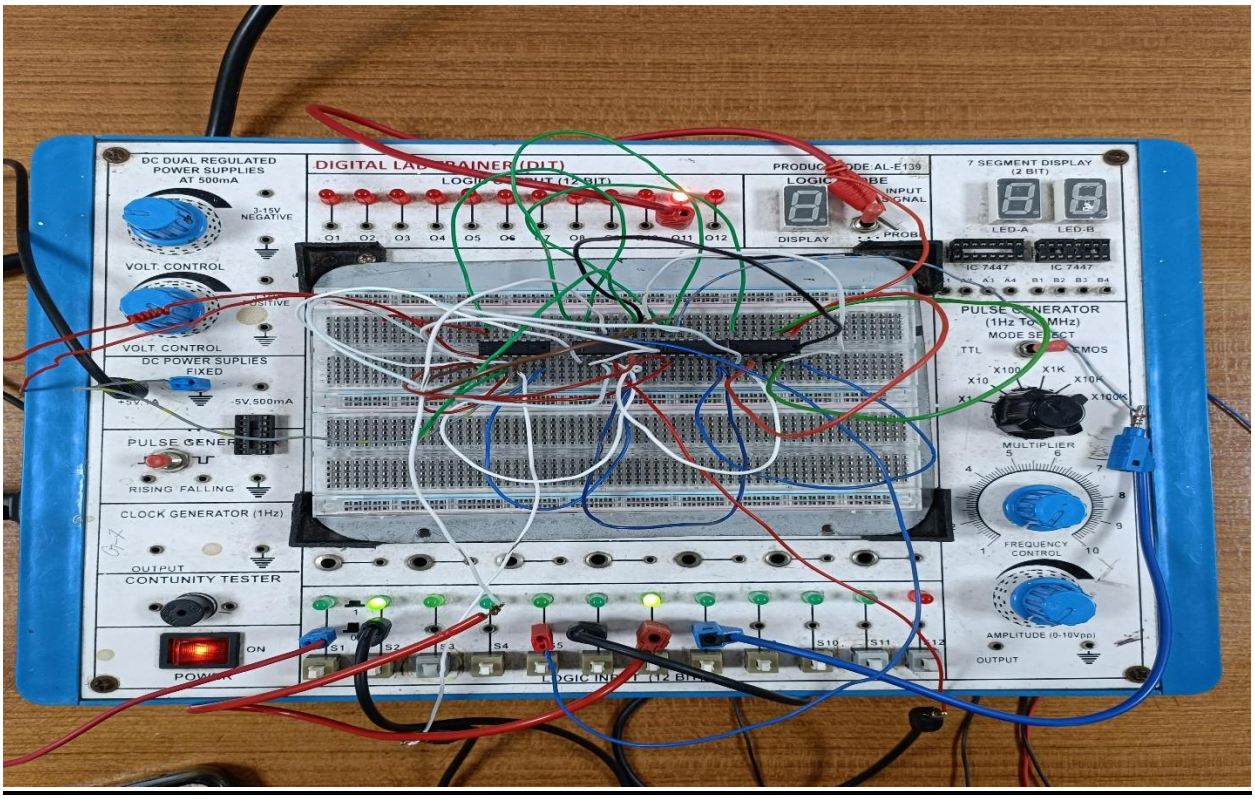
A3	A2	A1	A0	Y1	Y0
x	1	x	x	1	0
x	0	1	x	0	1
1	0	0	x	1	1
0	0	0	1	0	0

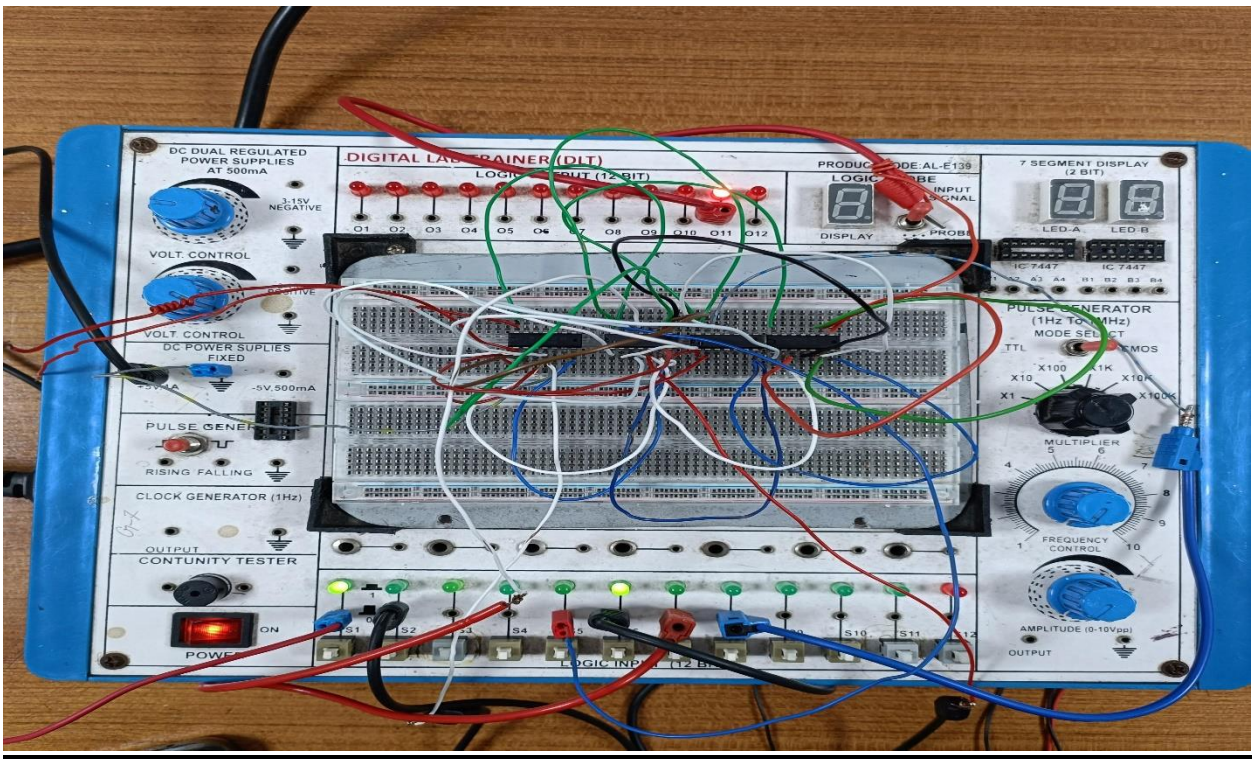
Apparatus:

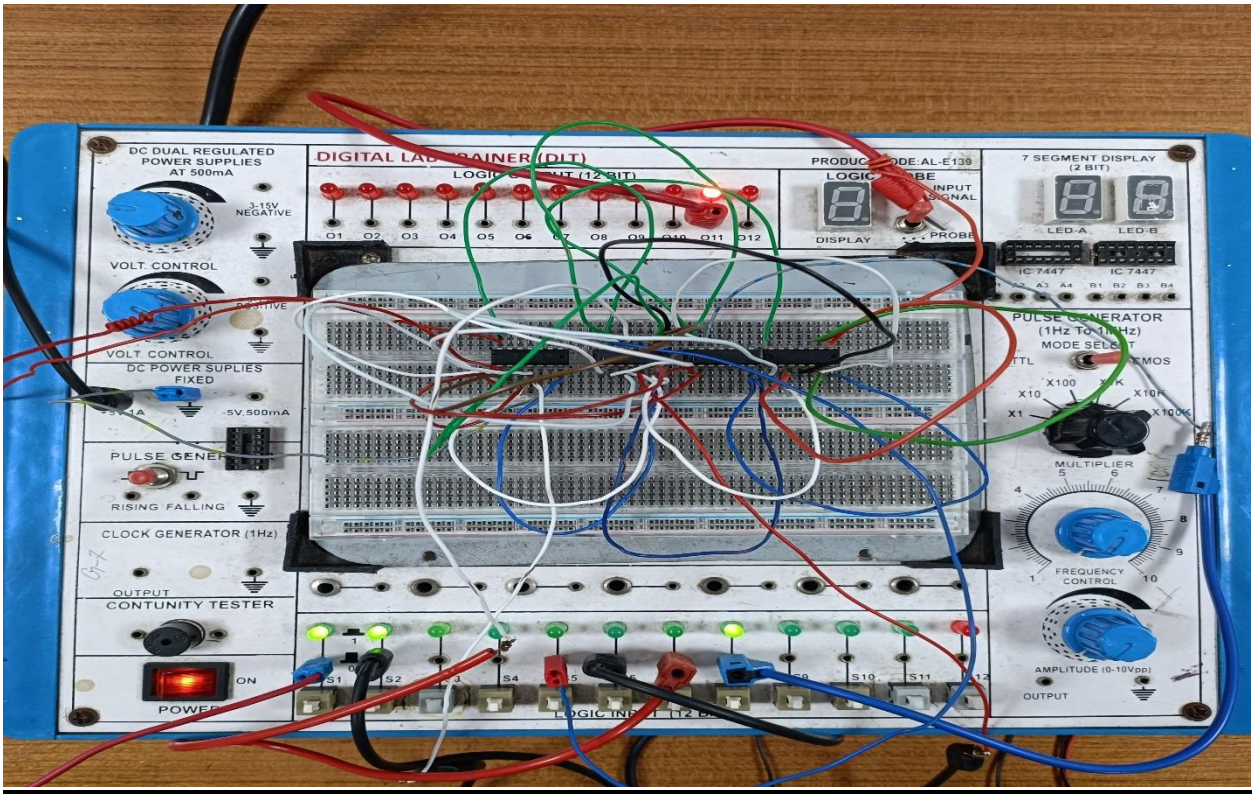
1. NOT Gate -	IC 7404	1[pcs]
2. AND Gate -	IC 7408	1[pcs]
3. OR Gate -	5 input OR	1[pcs]
	4 input OR	2[pcs]
	2 input OR	1[pcs]

Hardware Implementation:

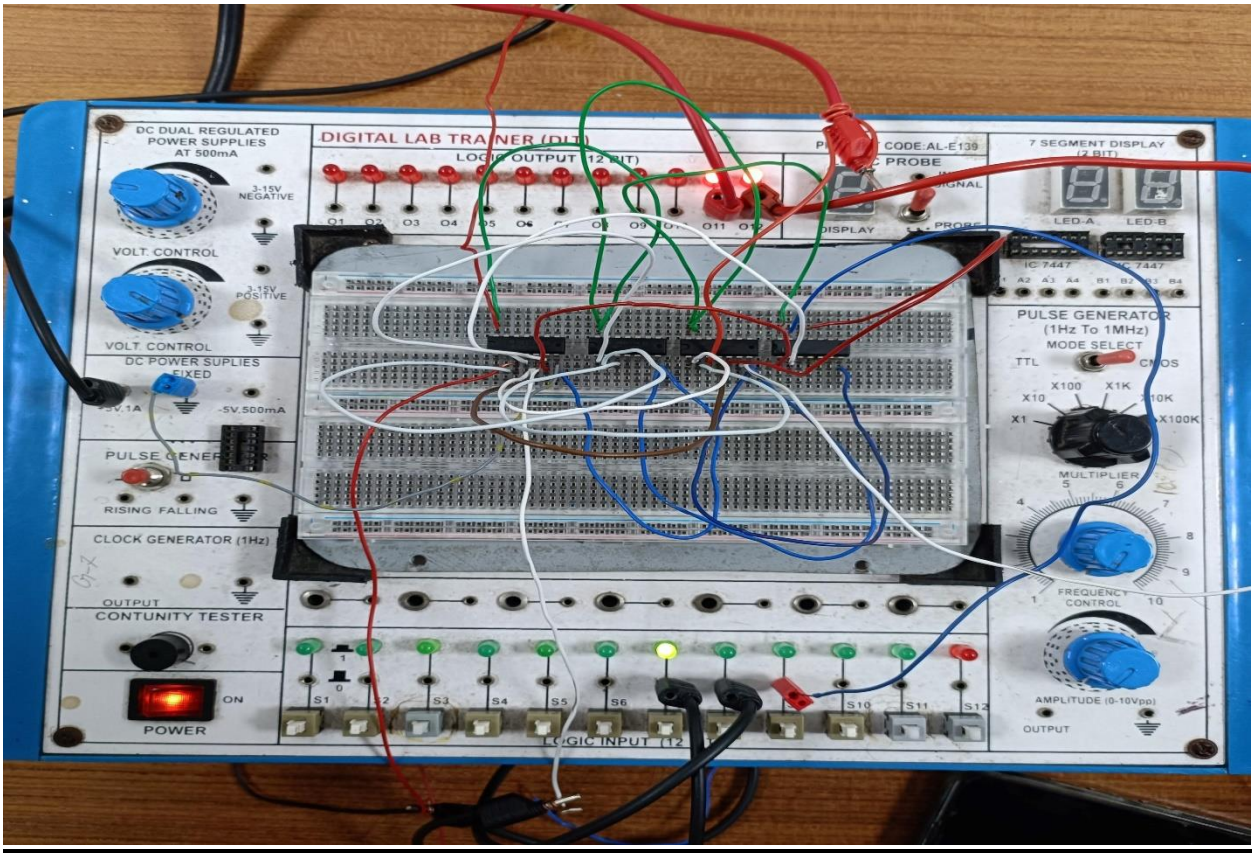
4 to 1 Multiplexer

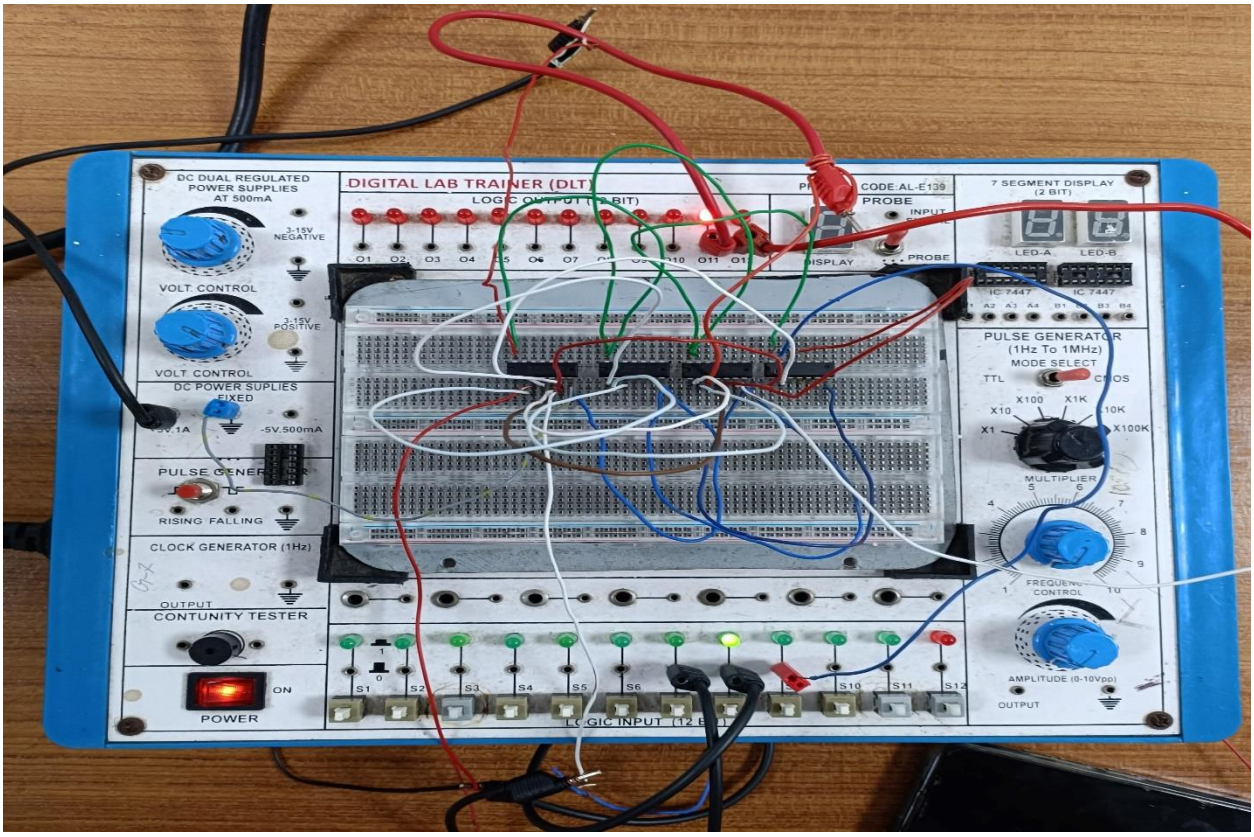


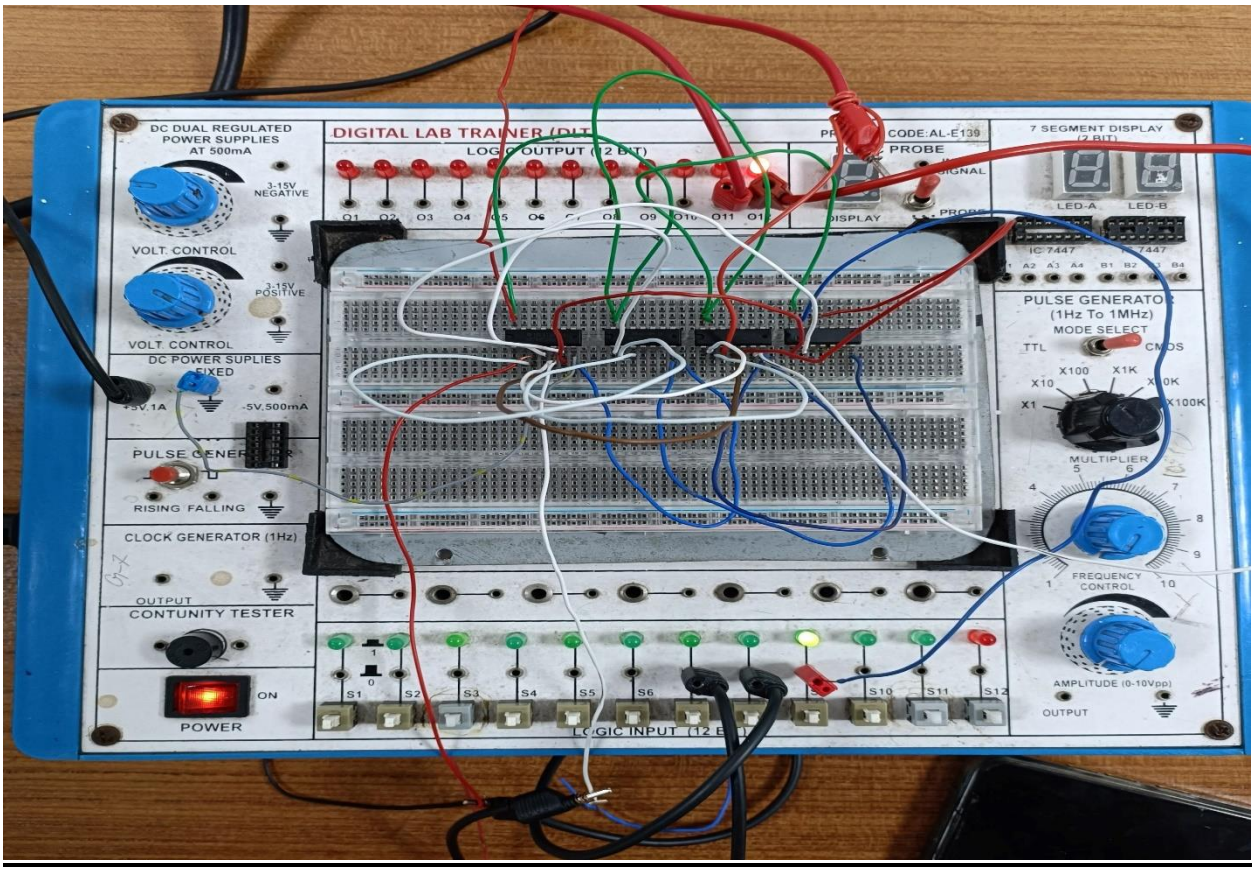




priority encoder

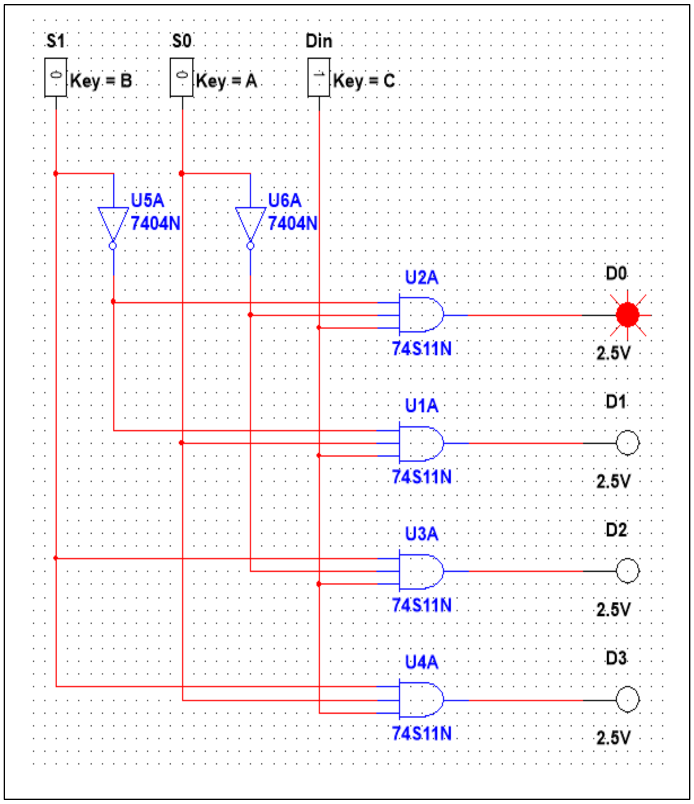


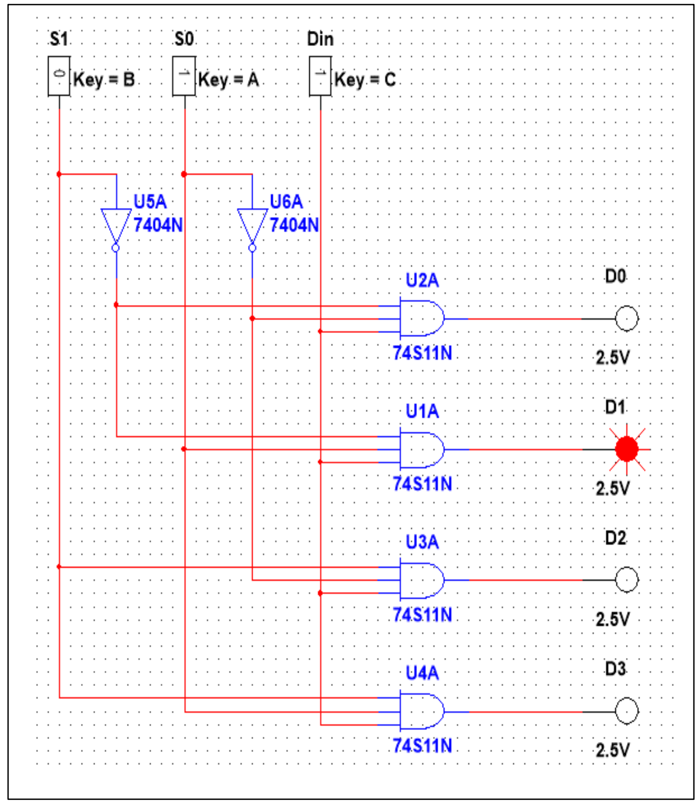

DC DUAL REGULATED POWER SUPPLIES AT 500mA
3-15V NEGATIVE
VOLT CONTROL
3-15V POSITIVE
VOLT CONTROL
DC POWER SUPLIES FIXED
+5V,1A
-5V,500mA
RISING FALLING
CLOCK GENERATOR (1Hz)
OUTPUT
CONTUNITY TESTER
ON
POWER
DIGITAL LAB TRAINER (DLT)
LOGIC OUTPUT (12 BIT)
CODE:AL-E139
PROBE
INPUT
DISPLAY
7 SEGMENT DISPLAY (2 BIT)
LED-A
LED-B
IC 7447
IC 7447
PULSE GENERATOR (1Hz To 1MHz)
MODE SELECT
TTL
CMOS
X100
X1K
X10
X10K
X1
X100K
MULTIPLIER
FREQUENCY CONTROL
AMPLITUDE (0-10Vpp)
OUTPUT
LOGIC INPUT (12 BIT)

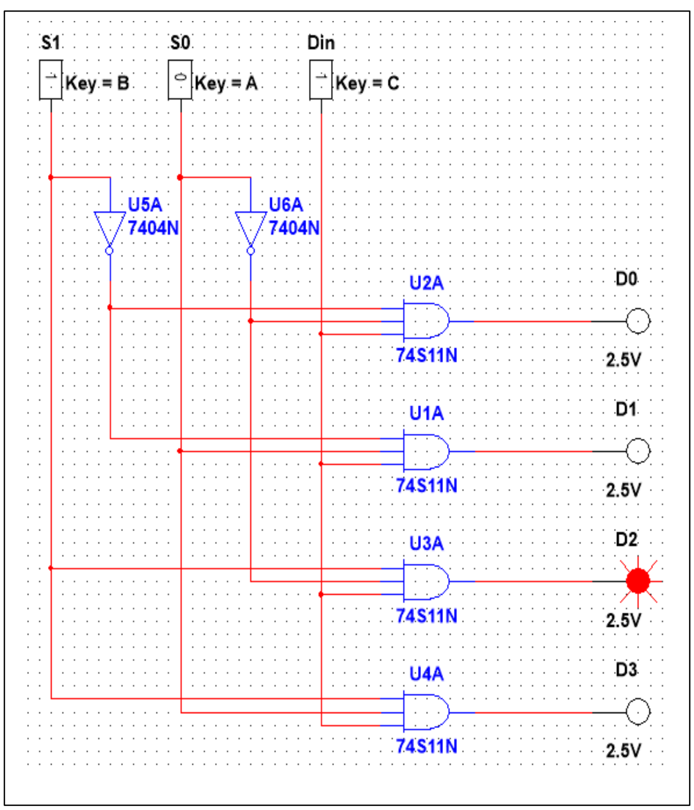

DC DUAL REGULATED POWER SUPPLIES AT 500mA
3-15V NEGATIVE
VOLT CONTROL
3-15V POSITIVE
VOLT CONTROL
DC POWER SUPLIES FIXED
+5V,1A
-5V,500mA
RISING FALLING
CLOCK GENERATOR (1Hz)
OUTPUT
CONTUNITY TESTER
ON
POWER
LOGIC OUTPUT (12 BIT)
PROBE
7 SEGMENT DISPLAY (2 BIT)
LED-A
LED-B
IC 7447
PULSE GENERATOR (1Hz To 1MHz)
MODE SELECT
TTL
CMOS
X100
X1K
X10
X1
X100K
MULTIPLIER
FREQUENCY CONTROL
AMPLITUDE (0-10Vpp)
OUTPUT

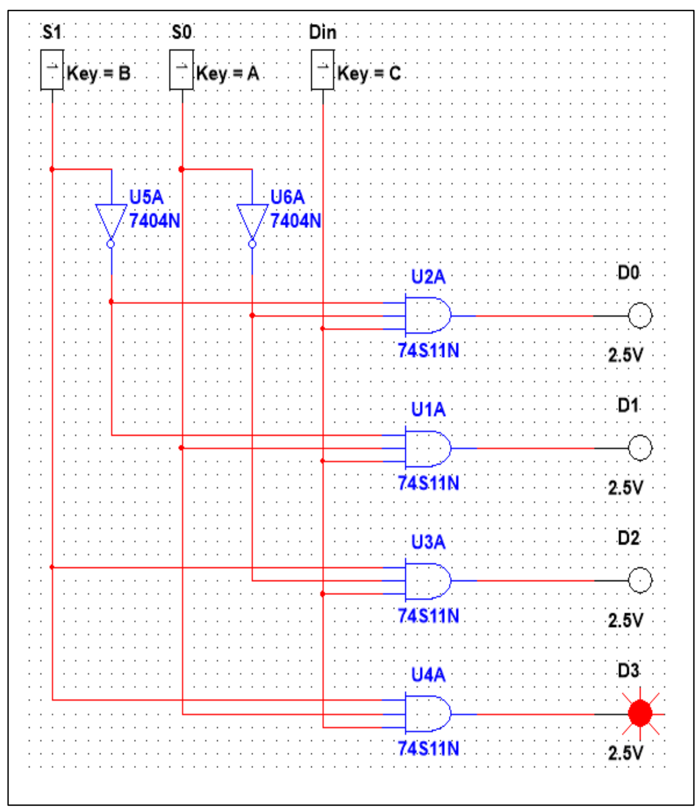
Simulation:

1-to-4 Demultiplexer:

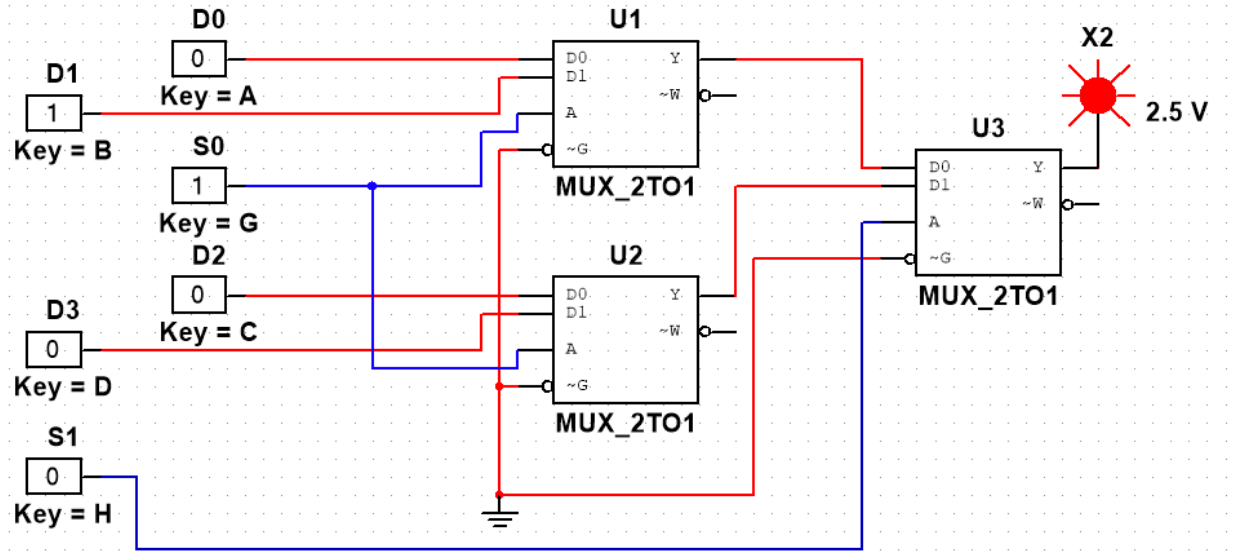


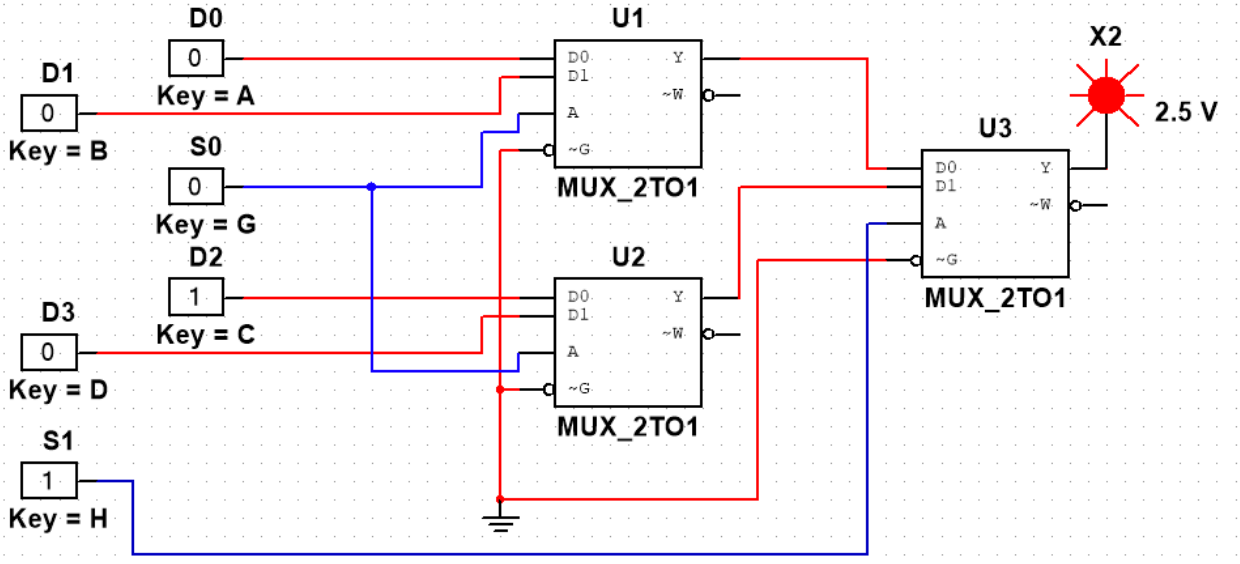




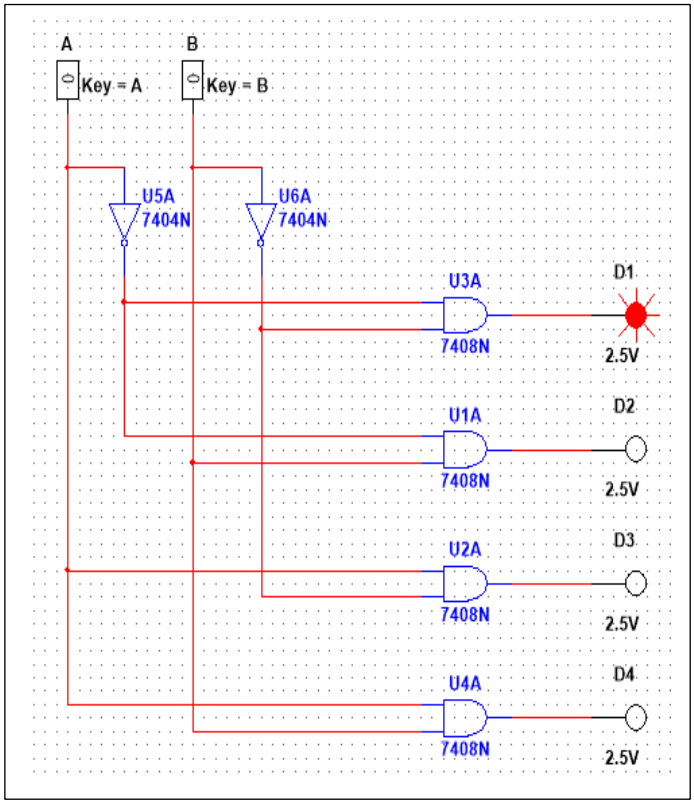


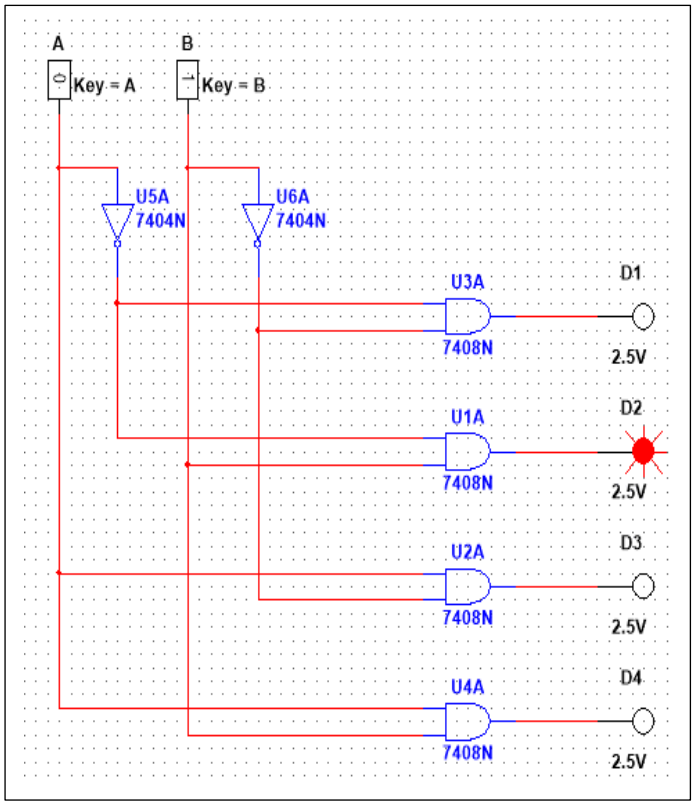
4 to 1 Multiplexer using 2 to 1 Multiplexer

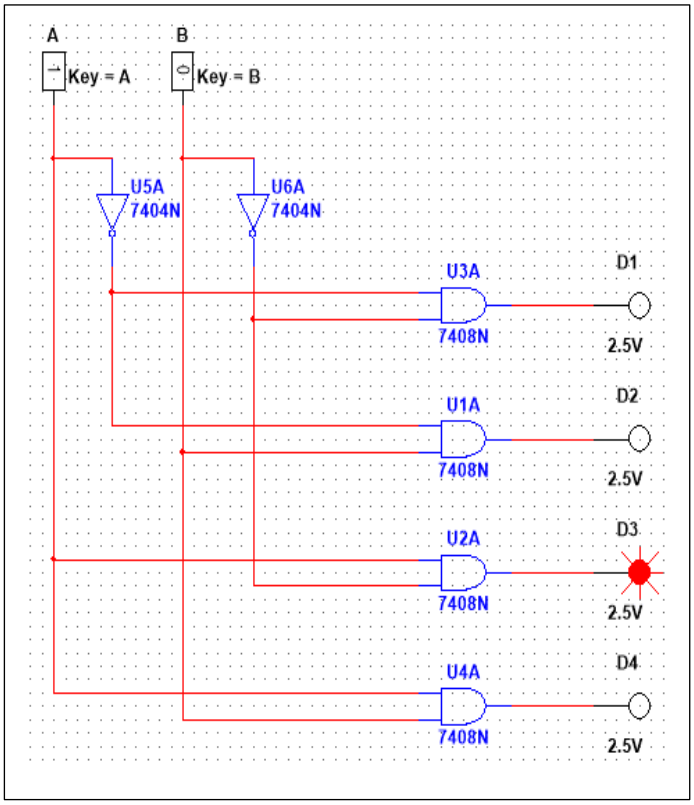


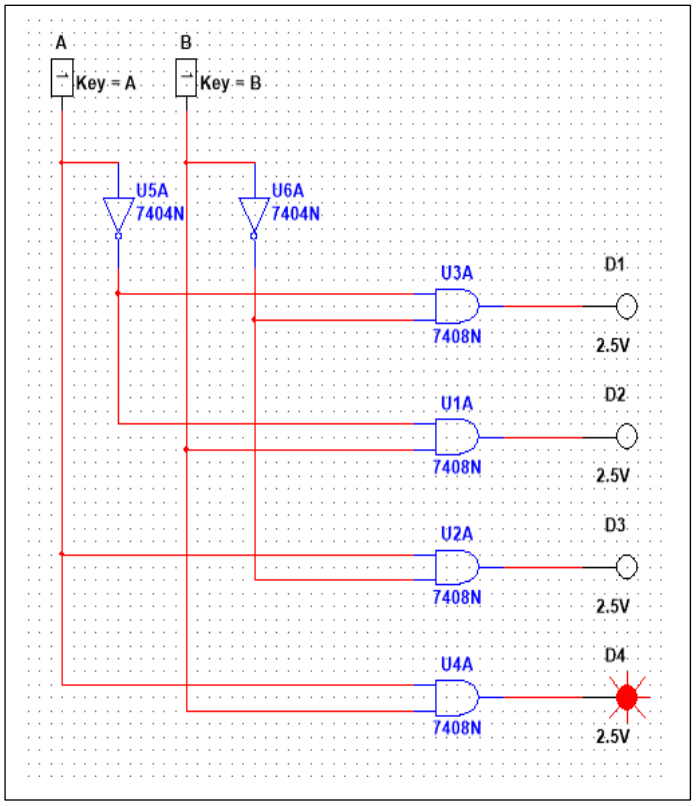


2-to-4-line decoder:









Decimal to BCD:

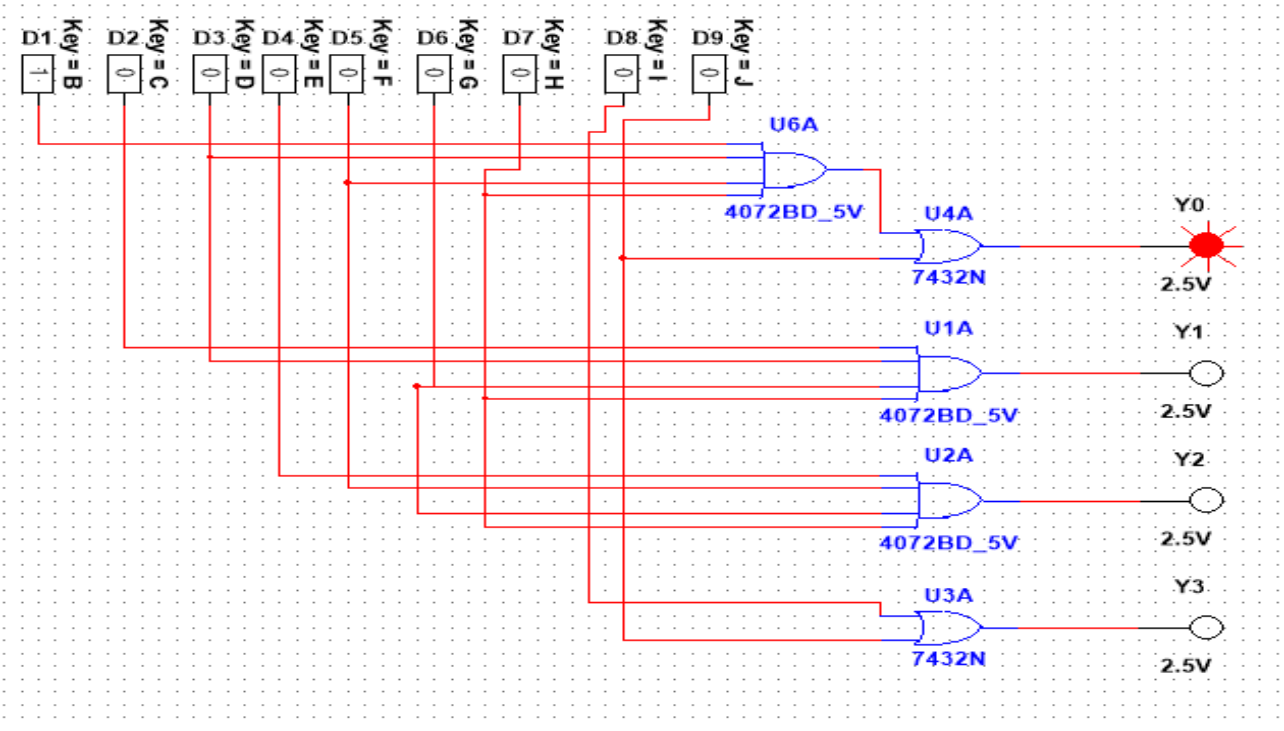


Figure D1

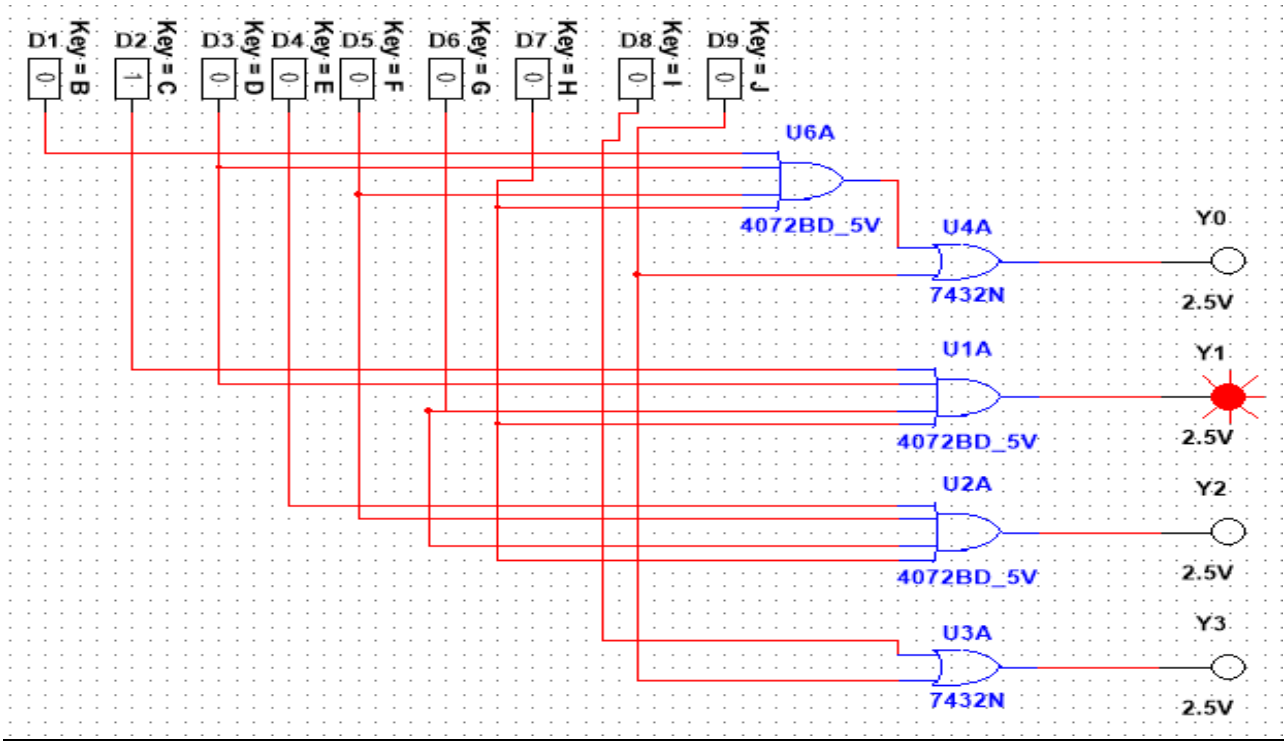


Figure D2

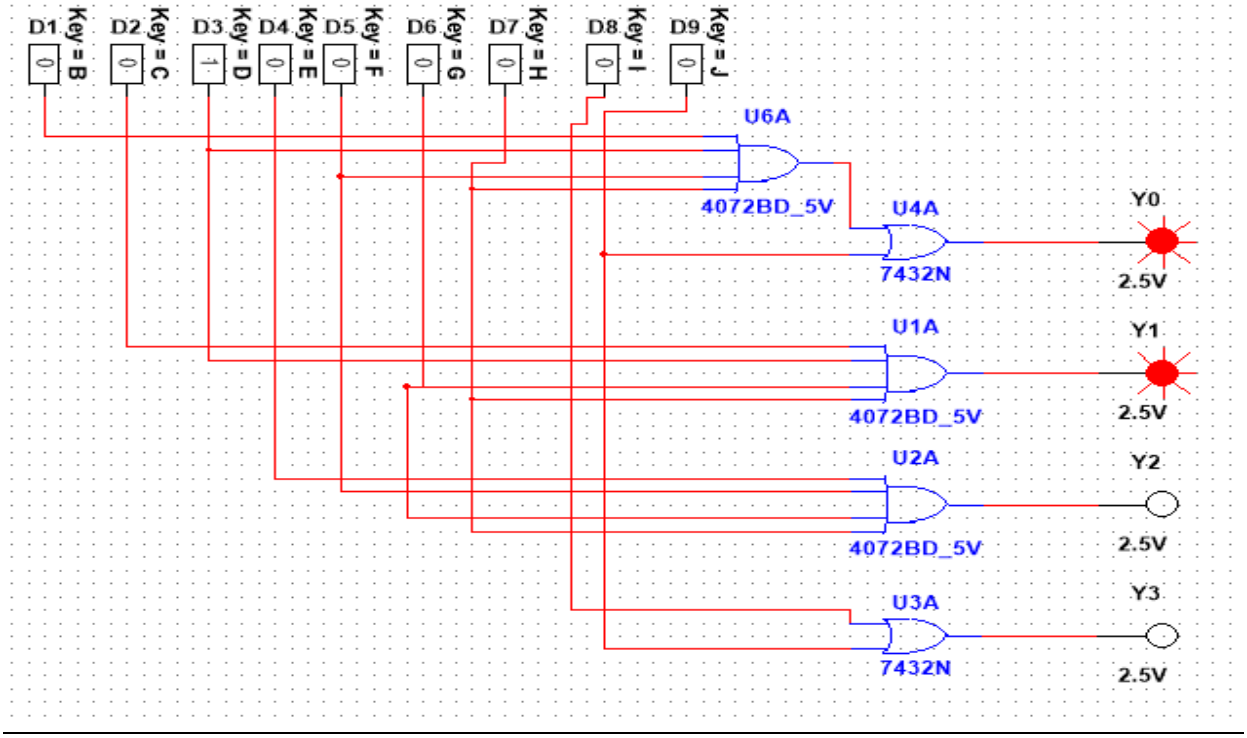


Figure D3

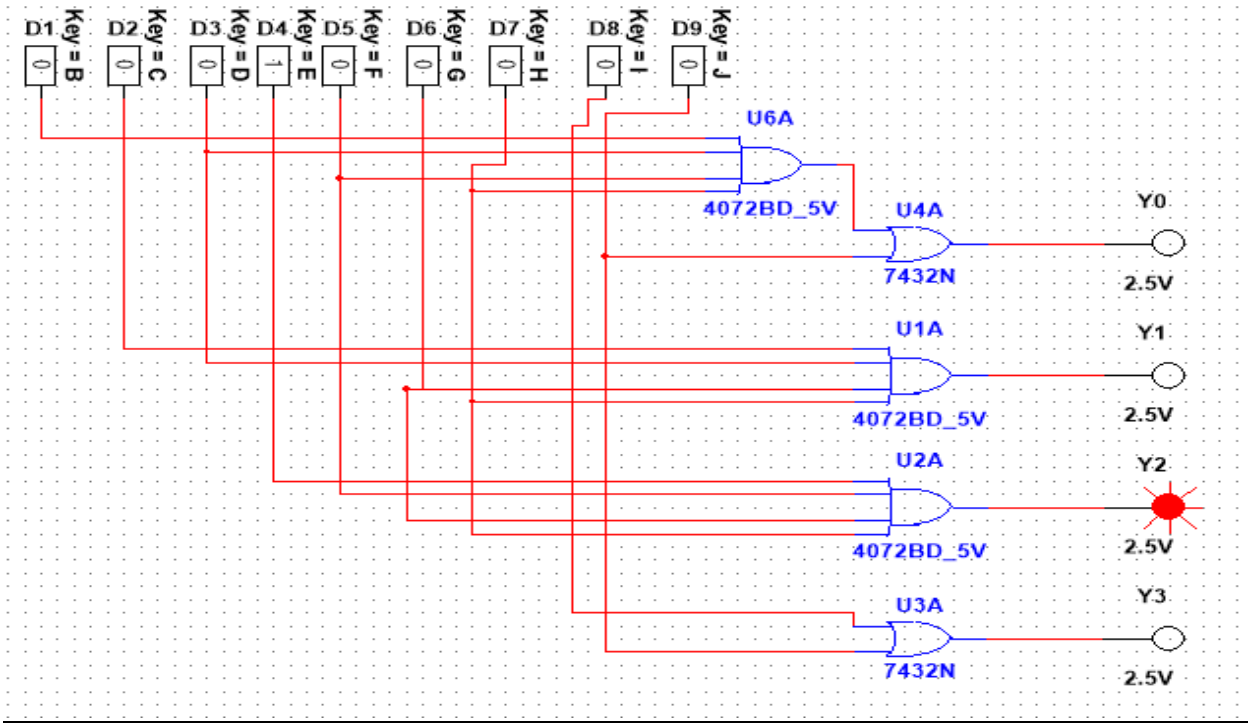


Figure D4

Fi

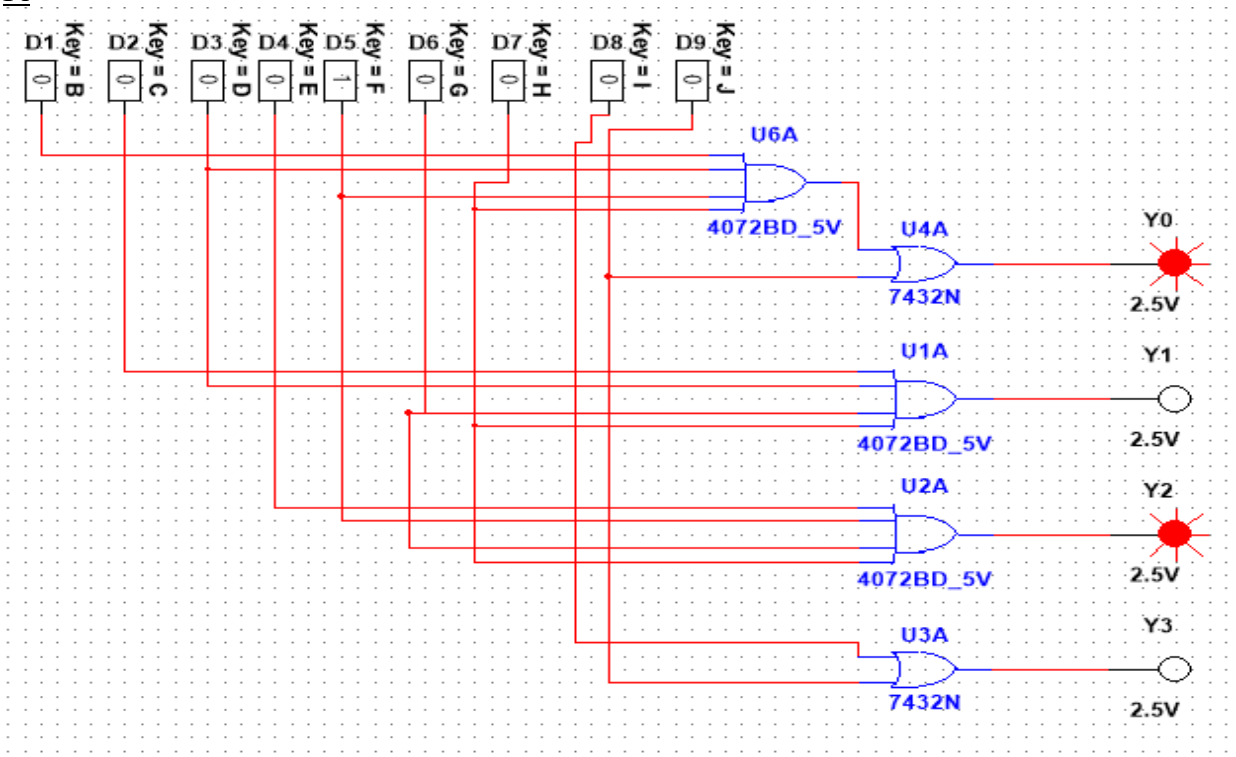


Figure D5

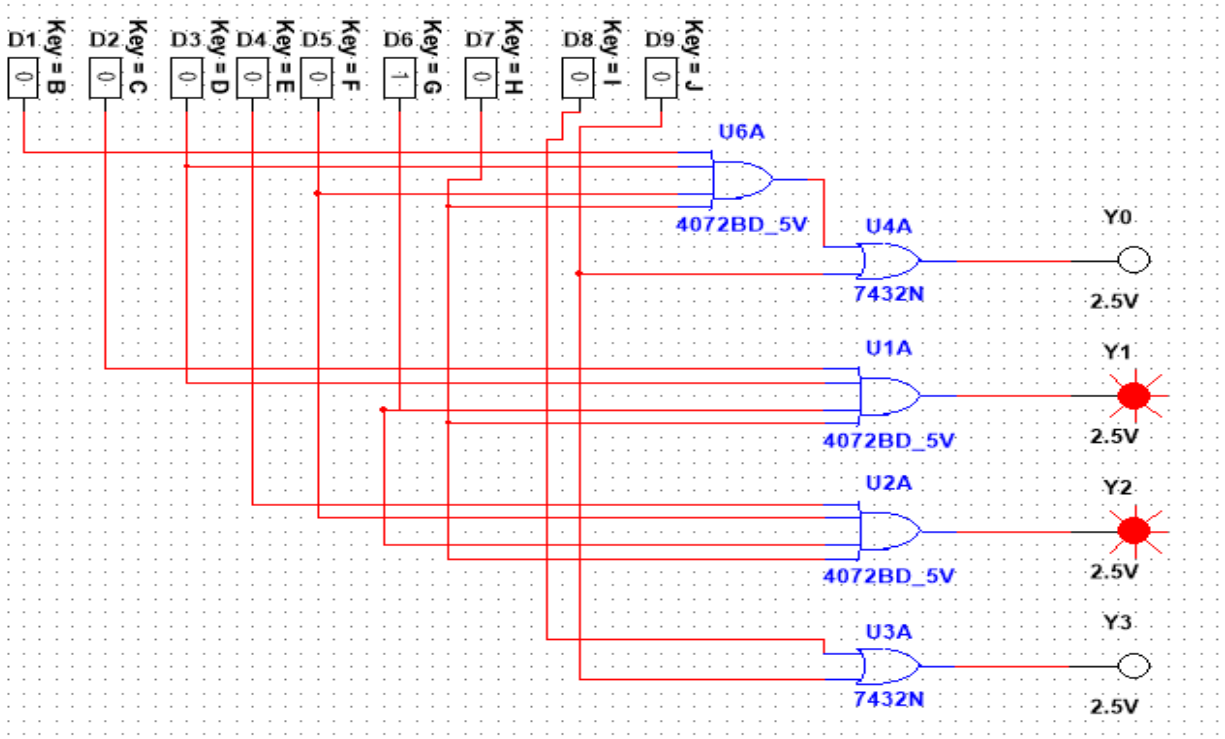


Figure D6

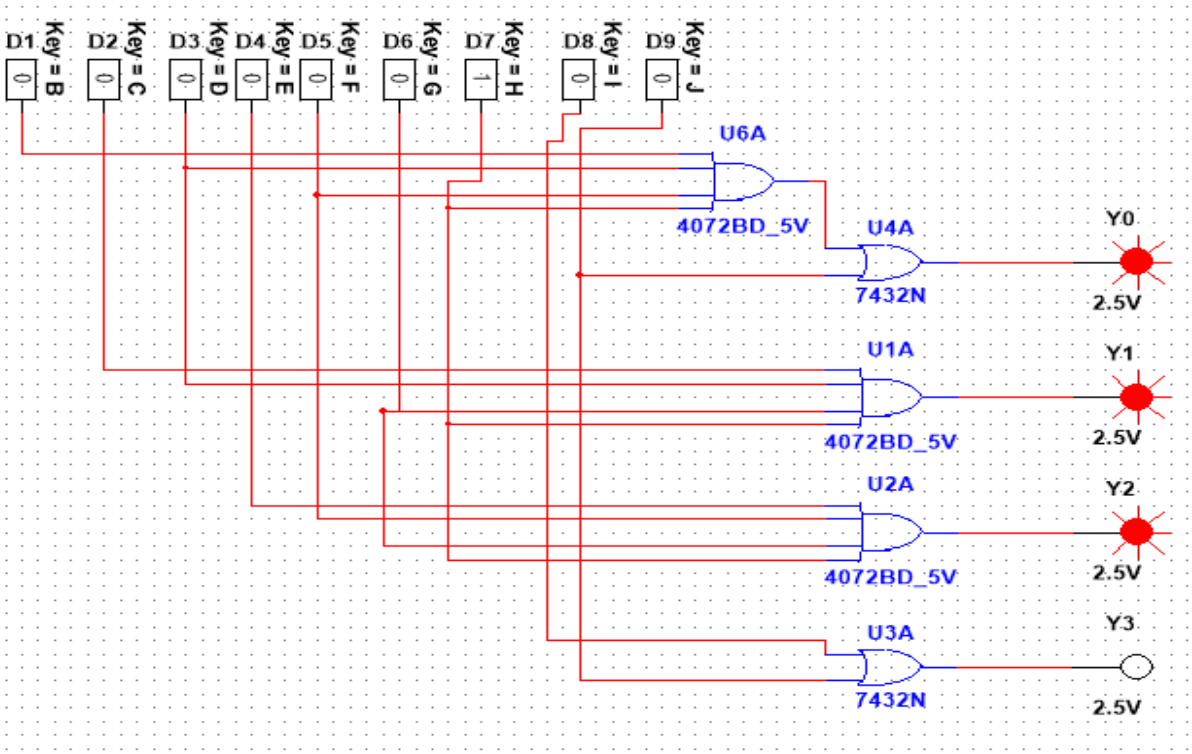


Figure D7

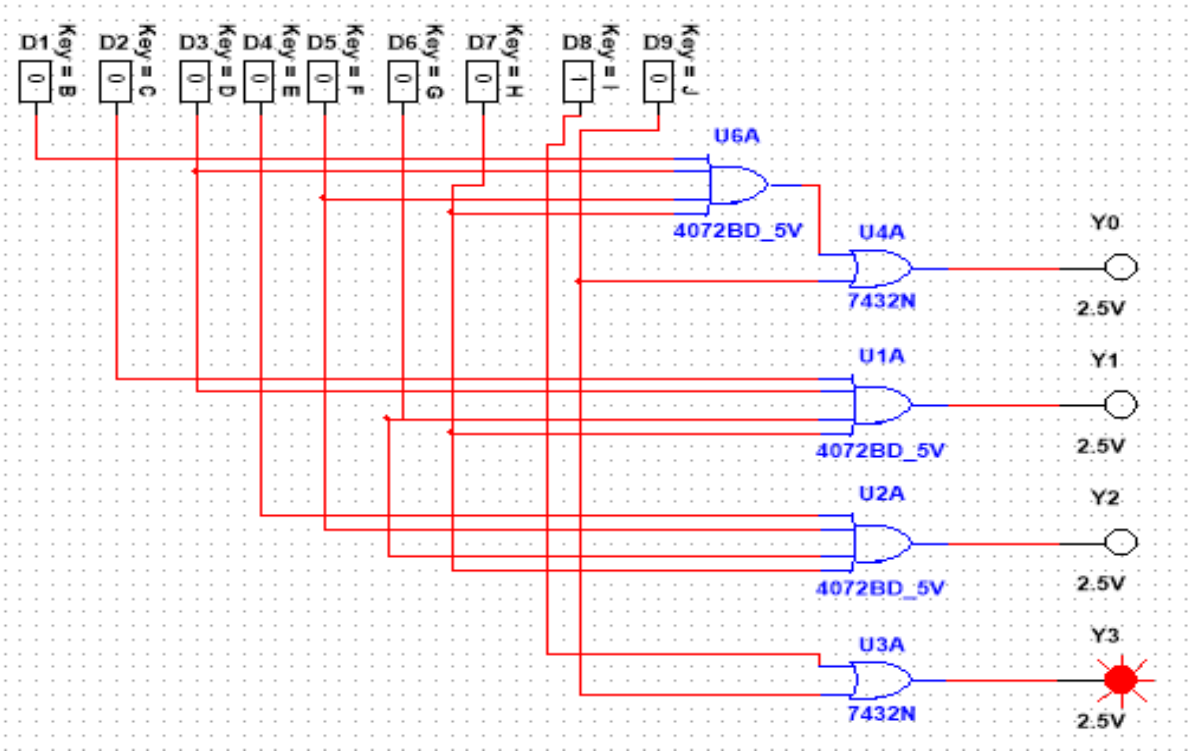


Figure D8

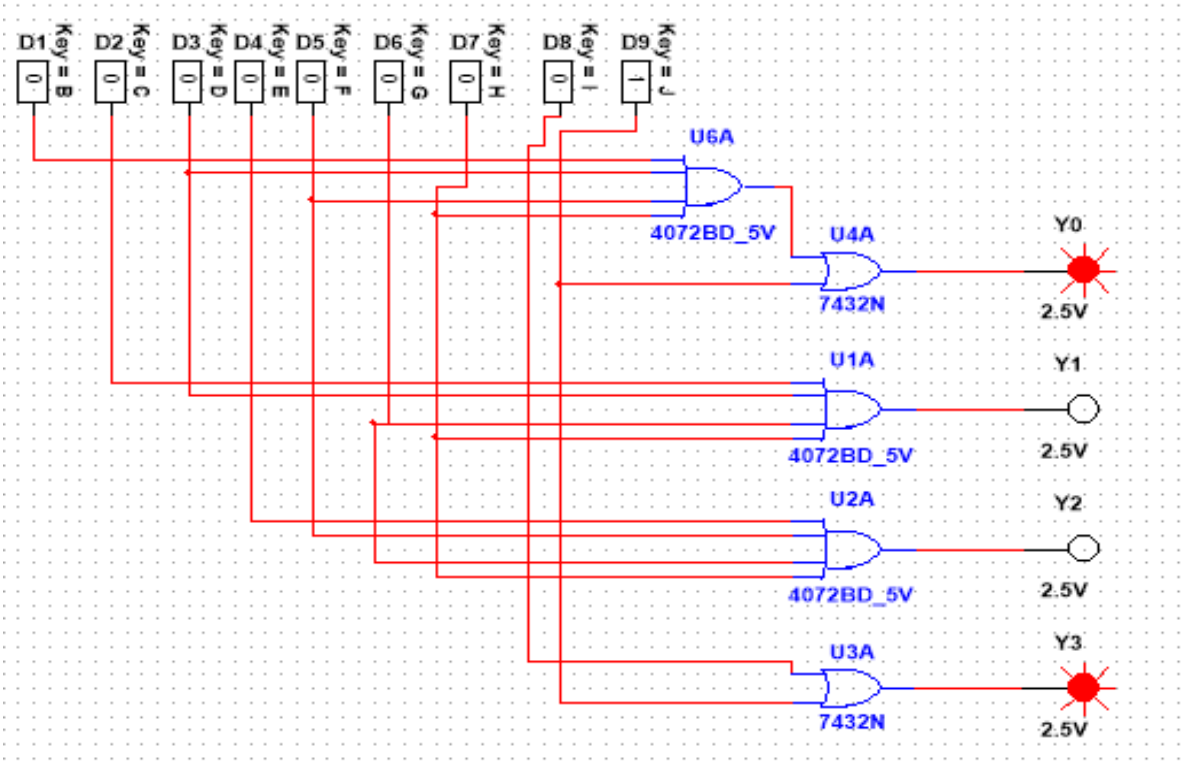


Figure D9

Results: The Simulation results matched all our theoretical truth table findings.

Conclusion:

During the experiment, mistakes could potentially occur, such as incorrect wiring connections with the ICs. However, to prevent any damage to the ICs, we took several precautions. Firstly, we thoroughly checked all the connections before powering on the circuit. Secondly, we made sure not to remove or insert ICs while the power was on. Lastly, we ensured that the voltage applied was within the safe range to avoid any potential harm to the ICs. By implementing these precautions, we aimed to minimize the risk of damaging the ICs and maintain a safe working environment.

Reference:

1. Thomas L. Floyd, “Digital Fundamentals,” available Edition, Prentice Hall International Inc.