# 

# Submitted By:

# 

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Contribution** |
| NUSHRAT JAHAN | **22-46149-1** |  |
| TRIDIB SARKAR | **22-46444-1** |  |
| S.M. MUJAHID SOUROV | **22-49679-3** |  |
| MD. ATIK ISHRAK SUJON | **22-46684-1** |  |
| MD. FAHIM MURSHED | **22-46695-1** |  |

Date of Submission: , 2023

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| **TOPICS** | **Page no.** |
| 1. **Cover Page** |  |
| **II. Table of Contents** |  |
| **Title** |  |
| **Abstract** |  |
| **Theory & Methodology** |  |
| **Apparatus** |  |
| **Experimental Procedure** |  |
| **Circuit Diagram** |  |
| **Images** |  |
| **IC Configuration** |  |
| **Simulation** |  |
| **Result & Discussion** |  |
| **References** |  |
|  |  |

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

**Abstract:** In this experiment we designed and implemented multiplexers (MUX) and demultiplexers (DeMUX) of different sizes using basic logic gates. We learned how to construct bigger multiplexer using smaller multiplexers and constructed 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 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.

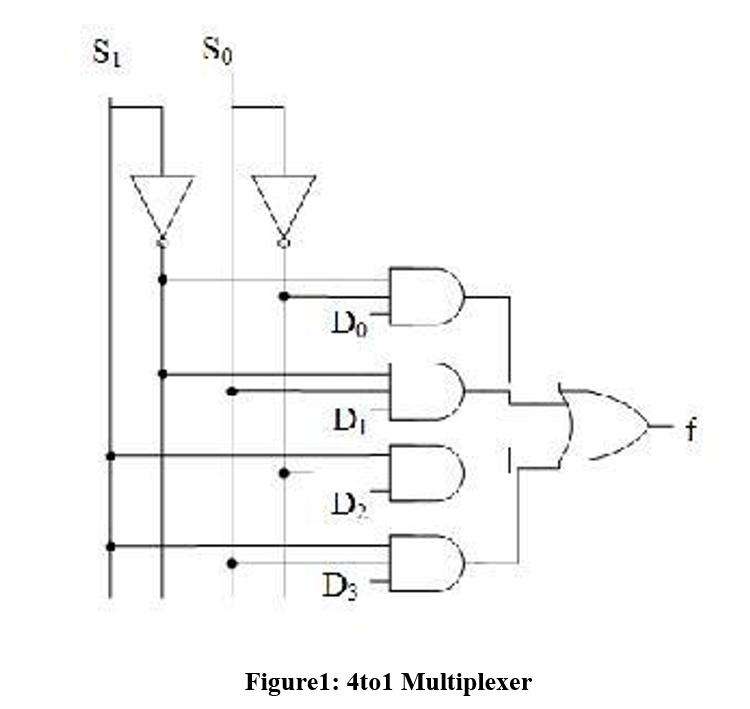
## Table:1

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

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

𝑓= 𝑆1̅𝑆0̅𝐷0+𝑆1̅𝑆0𝐷1+𝑆1𝑆0̅𝐷2+𝑆1𝑆0𝐷3 … (1)

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

****

**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 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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |

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

̅ … (3)

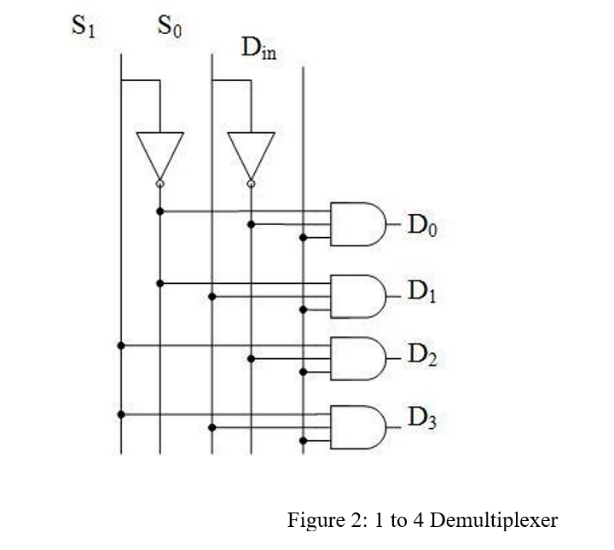
𝐷0=𝑆1̅𝑆0̅𝐷𝑖𝑛 … (2)

𝐷1=𝑆1̅𝑆0𝐷𝑖𝑛 … (3)

𝐷2=𝑆1𝑆0̅𝐷𝑖𝑛 … (4)

𝐷0=𝑆1̅𝑆0̅𝐷𝑖𝑛 … (5)

The circuit for 1to4 line demux is given below.



# 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.

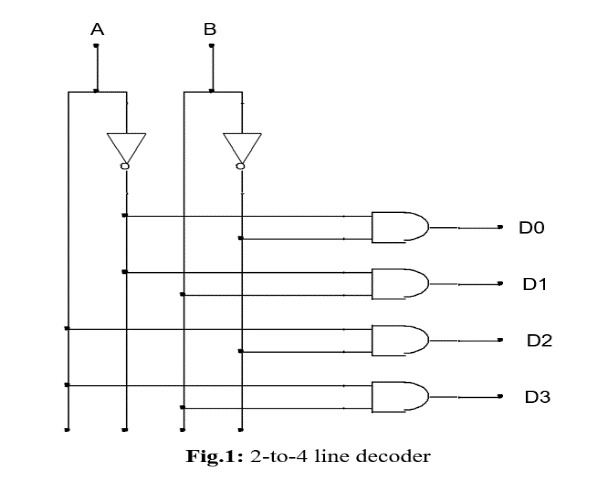
Encoding and decoding are very widely used ideas. They have applications in electronic circuits, software programs, medical devices, telecommunication and many others. In this experiment, a very basic 2-to-4 line decoder and a decimal to BCD encoder will be constructed.

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.

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 |





**Priority encoder:**

A priority encoder is a [circuit](http://en.wikipedia.org/wiki/Electronic_circuit) or [algorithm](http://en.wikipedia.org/wiki/Algorithm) that compresses multiple [binary](http://en.wikipedia.org/wiki/Binary_code) 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](http://en.wikipedia.org/wiki/Interrupt_request) 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.](http://en.wiktionary.org/wiki/precedence)

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

Truth table for this priority encoder is given below

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

Y0 = A3 + A1 A3’ A0’

Y1 = A3 + A2 A3’ A1’ A0’

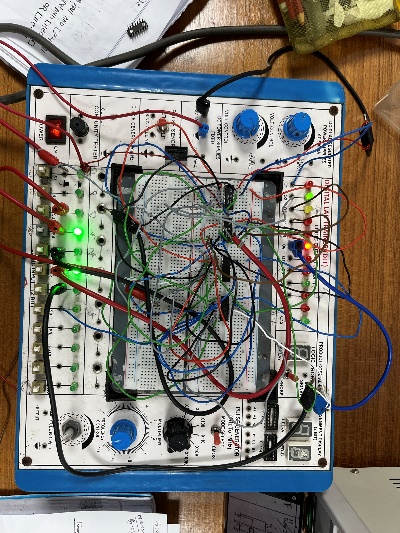
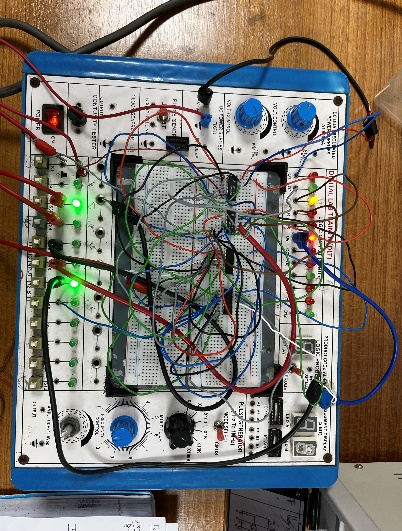
**Apparatus:**

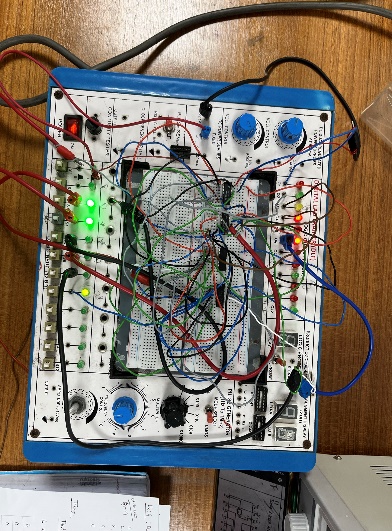
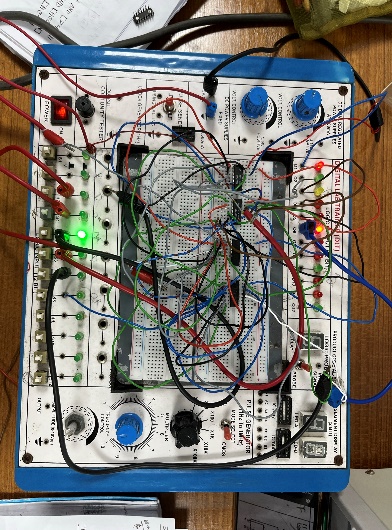
|  |  |
| --- | --- |
| NOT Gate - | IC 7404 |
| AND Gate - | IC 7408 |
| OR Gate - | 5 input OR |
|  | 4 input OR |
|  | 2 input OR |

**Experimental procedure:**

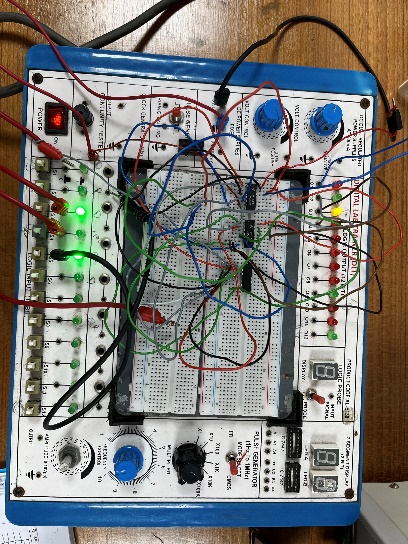
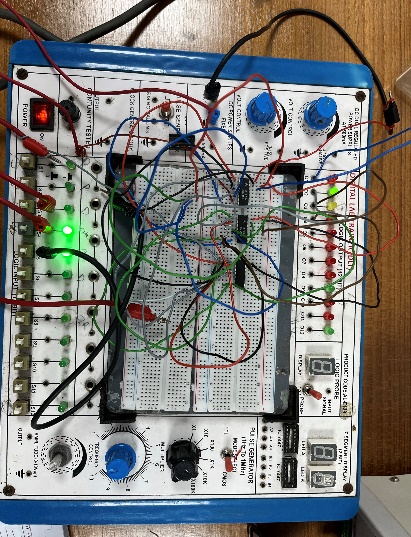
1. The circuit was connected according to the figures.
2. We connected the toggle switches on the trainer board to provide input signals to the circuits. The outputs were connected to the LEDs on the trainer board.
3. We applied the input signals and observed and noted the corresponding output signals.

**Simulation and Measurement:**

**Experimental circuit:**



# Figure1: 4to1 Multiplexer



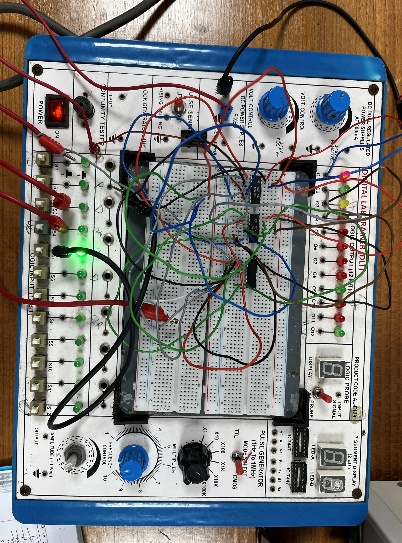
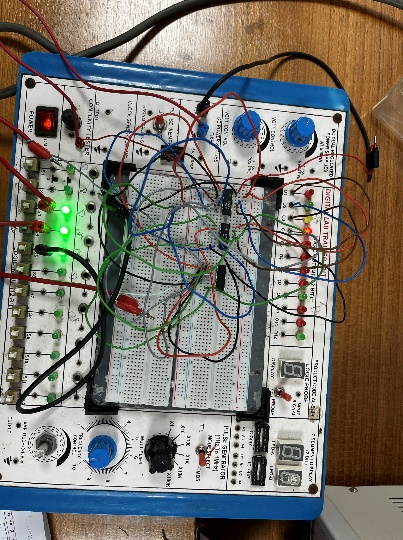
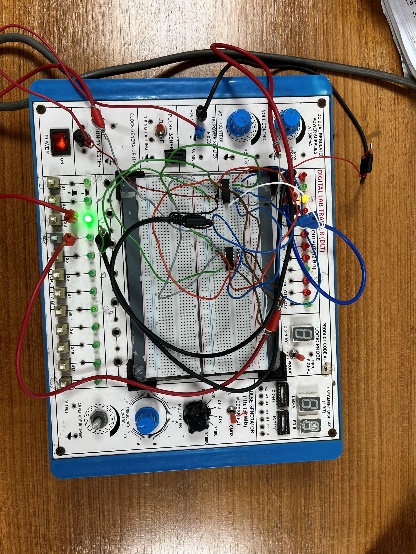
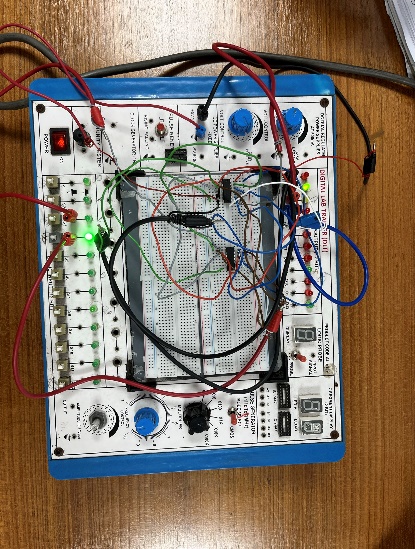
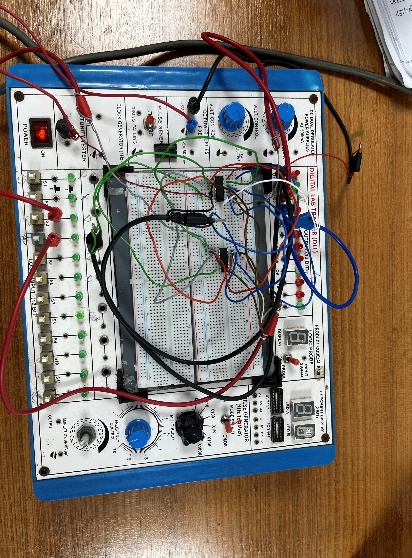
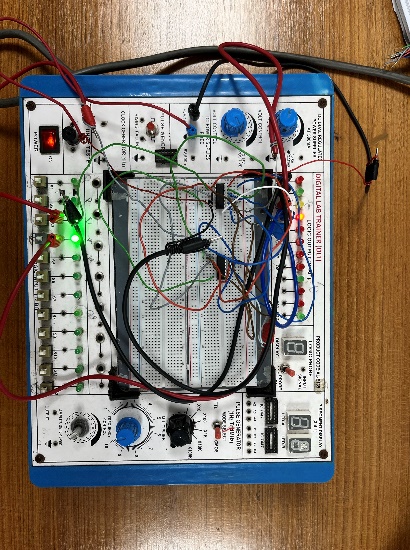
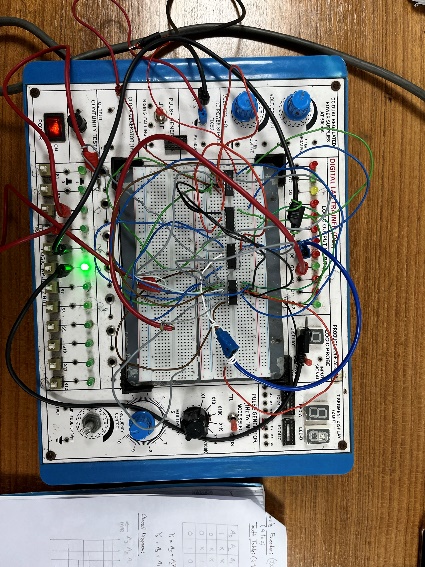
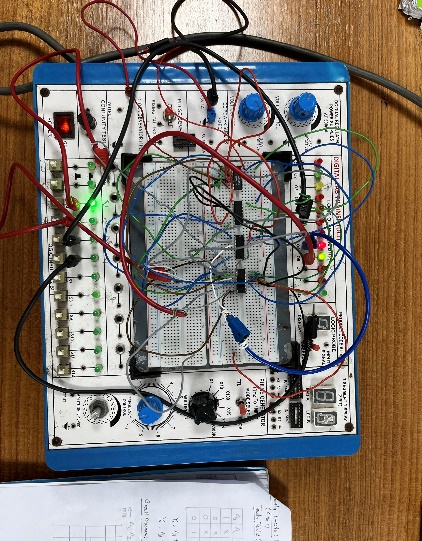


Figure 2: 1 to 4 Demultiplexer





Fig.3: 2-to-4 line decoder

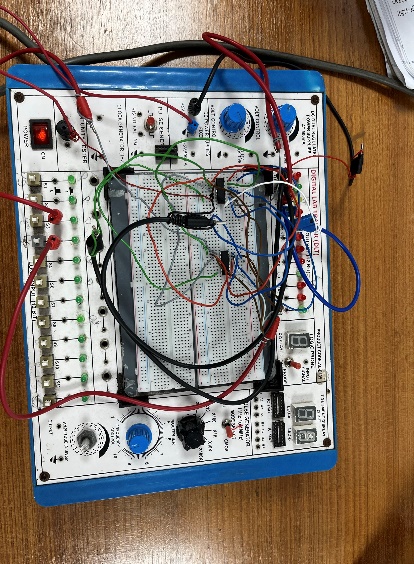
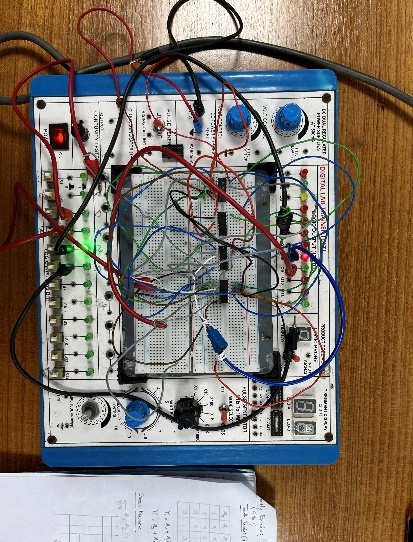


Fig .4: 4-to 2 priority encoder

**Result and discussion:** Inthis lab experiment, we successfully designed and implemented combinational circuits for multiplexers, demultiplexers, encoders and decoders using Multisim software. Despite challenges in integrating the components, we assembled the circuits on the breadboard without errors. Simulation results match the expected truth tables, validating our designs. Designing these components deepened our understanding of digital logics circuitry. Through the hands-on experience and simulations, we gained proficiency in truth table design and verification. We can now confidently explain the functionalities of multiplexers, demultiplexers, encoders and decoders.

Overall, this lab exercise enhanced or knowledge and skills in digital logic design. Successful verification of truth table conclusions through simulations validated the accuracy of our work, solidifying our comprehension of these essential components in digital circuits.

**Reference:**

* <http://www.tutorialspoint.com/computer_logical_organization/combinational_circuits.htm>