

**Faculty of Engineering and Technology Electrical and Computer Engineering Department**

**DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY ENCS2110**

**Experiment No. 3**

Encoders, Decoders, Multiplexers, and Demultiplexers

**Prepared by:**

**1221480 layal abed**

**Partners:**

**Rema Qashoo 1221009**

**Aya Abu Sneineh 1221414**

**Instructor: Abualseoud Hanani**

**Teaching assistant:Hanan**

**Section : 2**

**Date: 15-7-2024**

# 

# Abstract

At the outset of this experiment, we revisited our understanding of Encoders, Decoders, Multiplexers, and Demultiplexers, delving into their block diagrams, circuits, and truth tables, as well as the implementation of Boolean functions using these components.

In the hands-on portion, we built each of these components using basic gates and then assembled them using their respective blocks. Furthermore, we implemented a logic function utilizing a multiplexer.

This experiment has significantly enhanced my comprehension of Encoders, Decoders, Multiplexers, and Demultiplexers, as well as my practical skills in working with them. I am now proficient in constructing these components both with basic gates and using their blocks.

.

**Table of Contents**

[Abstract I](#_Toc26926)

[Table of Contents II](#_Toc29205)

[Table of Figures III](#_Toc30957)

[List of Tables IV](#_Toc1520)

[1.Theory 1](#_Toc10160)

[1.1 Decoder 1](#_Toc8130)

[1.1.1 2 to 4 Decoder: 1](#_Toc14472)

[Two to four Decoder using basic gates: 2](#_Toc24693)

[1.2 Encoder: 3](#_Toc29285)

[1.3 Multiplexer: 7](#_Toc15901)

[1.4 De-Multiplexer: 9](#_Toc14923)

[2.Procedure and Discussion 11](#_Toc15340)

[2.1 constructing a 4-to-2 Encoder With Basic Gates: 11](#_Toc22524)

[2.1.1connection: 11](#_Toc22181)

[2.1.2 Results : 12](#_Toc265)

[2.1.3Discussion 13](#_Toc29169)

[2.2 Constructing 9-to-4-Line Encoder with TTLIC 14](#_Toc25970)

[2.2.1 connection 14](#_Toc13560)

[2.2.2 Results: 15](#_Toc4664)

[2.2.3 Discussion : 16](#_Toc31024)

[2.3Constructing 2-to-4 Line Decoder with basic gates 17](#_Toc24976)

[2.3.1Connection: 17](#_Toc17478)

[2.3.2Results: 18](#_Toc11671)

[2.3.3Discussion: 19](#_Toc20091)

[2.4Constructing 4-to-10 Line Decoder with TTLIC: 19](#_Toc25013)

[2.4.1Connection: 19](#_Toc24836)

[2.4.2Results: 21](#_Toc10260)

[2.4.3Discussion: 21](#_Toc29723)

[2.5Constructing 2-to-1 Line Multiplexer with Basic Gates: 22](#_Toc7125)

[2.5.1Connection: 22](#_Toc29413)

[2.5.2Results: 23](#_Toc28140)

[2.5.3Discussion: 24](#_Toc31588)

[2.6Constructing 8-to-1 Line Multiplexer with IC 24](#_Toc6594)

[2.6.1Connection 24](#_Toc25350)

[2.6.2 Results 25](#_Toc28692)

[2.6.3 Discussion : 26](#_Toc19597)

[2.7Constructing 1-to-2 Line Demultiplexer with Basic Logic Gates: 26](#_Toc12714)

[2.7.1Connections: 26](#_Toc17000)

[2.7.2Results: 27](#_Toc216)

[2.7.3Discussion: 27](#_Toc8382)

[2.8Constructing 1-to-8-Line Demultiplexer with CMOS IC 28](#_Toc26593)

[2.8.1 Connection: 28](#_Toc13476)

[2.8.2 Results 29](#_Toc25538)

[2.8.3 Discussion : 30](#_Toc25193)

[Conclusion: 30](#_Toc7649)

[References: 31](#_Toc29402)

**Table of Figures**

[Figure 1 :1.1.1: General decoder block diagram 1](#_Toc26651)

[Figure 2 : 1.1.1.1: 2 to 4 decoder with enable 2](#_Toc2123)

[Figure 3 : 1.1.1.2: 2 to 4 Decoder circuit 3](#_Toc5194)

[Figure 4 : 1.2.1 : General encoder block diagram 4](#_Toc1967)

[Figure 5 : 1.2.1.1 : 4 to 2 Encoder 4](#_Toc30548)

[Figure 6 : 1.2.1.2 : 4 to 2 Encoder circuit 5](#_Toc24769)

[Figure 7 : 1.2.2.1: 8 to 3 priority Encoder 6](#_Toc7574)

[Figure 8 : 1.3.1: 4 to 1 Mux block diagram 7](#_Toc25452)

[Figure 9 : 3.1.2: 4 to 1 Multiplexer circuit 8](#_Toc15323)

[Figure 10 : 1.4.1: 1 to 4 De-Mux block diagram 9](#_Toc32127)

[Figure 11 : 1.4.2: 1 to 4 De-Multiplexer 10](#_Toc22053)

[Figure 12 : 2.1.1.1: wiring diagram of 4-to-2line Encoder 11](#_Toc14234)

[Figure 13 : 2.1.1.2 : connection of 4-to-2 Encoder with basic gates 12](#_Toc17323)

[Figure 14 : 2.2.1.1 : 74147 BCD Priority Encoder 14](#_Toc31054)

[Figure 15 : 2.2.1.2 : connection 9-to-4 Line Encoder 15](#_Toc13342)

[Figure 16 : 2.3.1.1: 2-t-4 decoder 17](#_Toc22656)

[Figure 17 : 2.3.1.2: connection 2-to-4 Line Decoder 18](#_Toc6704)

[Figure 18 : 2.4.1.1: 4-to-10 Line Decoder 19](#_Toc30734)

[Figure 19 : 2.4.1.2 : conection 4-to-10 Line Decoder 20](#_Toc8023)

[Figure 20 : 2.5.1.1: 2-to-1 Multiplexer 22](#_Toc14774)

[Figure 21 : 2.5.1.2 : connection 2-to-1 Line Multiplexer with basic gates 23](#_Toc1230)

[Figure 22 : 2.6.1.1 : 8-to-1 Mux 24](#_Toc18227)

[Figure 23 : 2.6.1.1 : connection 8-to-1 Line Multiplexer with IC 25](#_Toc5598)

[Figure 24 : 3.17 : 1-to-2 Demultiplexer 26](#_Toc13992)

[Figure 25 : 2.7.1.2 : connection 1-to-2 Line Demultiplexer with basic gates 27](#_Toc3931)

[Figure 26 : 2.8.1.1: 1-to-8 Demultiplexer 28](#_Toc20034)

[Figure 27 :2.8.1.2: connection 1-to-8-Line Demultiplexer with CMOS IC 29](#_Toc8305)

# List of Tables

[Table 1 : 1.1.1.1: Truth table of 2 to 4 decoder with enable 2](#_Toc29857)

[Table 2 : 1.2.1.1: Truth table of 4 to 2 encoder 5](#_Toc17397)

[Table 3 : 1.2.2.1: 8 to 3 priority Encoder truth table 6](#_Toc14256)

[Table 4 : 1.3.1: Truth table of 4 to 1 Multiplexer 7](#_Toc18911)

[Table 5 : 1.4.1: Truth table of 1 to 4 De-Multiplexer 9](#_Toc16179)

[Table 6 : 2.1.1.1: Data for 4-t-2 Encoder with basic gates 12](#_Toc28398)

[Table 7 : 2.2.2.1: the data of priority Encoder 15](#_Toc24301)

[Table 8 : 2.3.2.1: 2-to-4 line Decoder with basic gates 18](#_Toc32039)

[Table 9 : 2.4.2.1 : Data for 4-to-10 Line Decoder 21](#_Toc18984)

[Table 10 : 2.5.2.1 : 2-to-1 Line Multiplexer with Basic Gates 23](#_Toc9287)

[Table 11 : 2.6.2.1 : Data for 8-to-1 Line Multiplexer 25](#_Toc8617)

[Table 12 : 2.7.2.1 : Data for 1-to-2 Line DeMux 27](#_Toc5974)

[Table 13 : 2.8.2.1 : data for 1-to-8 DeMux 29](#_Toc7651)

# 1.Theory

**1.1** **Decoder :**

Decoders are a combinational circuit that changes the binary information into 2N output lines, the decoder is an example of multiple-input multiple-output combinational logic circuit device, This binary information passed to decoder as N input lines and then it converted to the 2N bit code as output , one of these output active(1) and the other is inactive (0) , the outputs of the decoder is the minterms of N input variables , so the decoder is the reverse of the Encoder.

This the Block diagram of the Decoder:

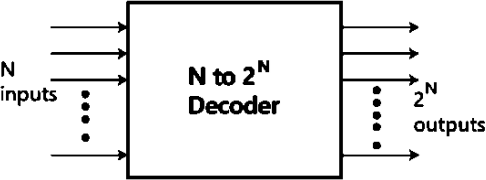


Figure 1:1.1.1: General decoder block diagram

### 1.1.1 2 to 4 Decoder:

2 to 4 decoder has two inputs A1 and A0 and four outputs Y3, Y2, Y1 and Y0 and its block diagram shown in this figure:

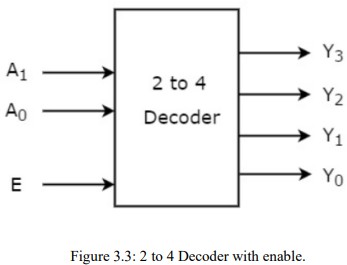


Figure 2: 1.1.1.1: 2 to 4 decoder with enable

When the Enable is "1" only one of these output will be "1" , and this the truth table of two to four decoder:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Enable | Inputs | | Outputs | | | |
| E | A1 | A0 | Y3 | Y2 | Y1 | Y0 |
| 0 | x | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Table 1: 1.1.1.1: Truth table of 2 to 4 decoder with enable

**Two to four Decoder using basic gates:**

As well as each output have one product term , so in the 2 -4 decoder there are four product terms , so we need an AND gates for each product and two inverters, the following figure is the circuit diagram of 2 to 4 decoder

1.2 Encoder:

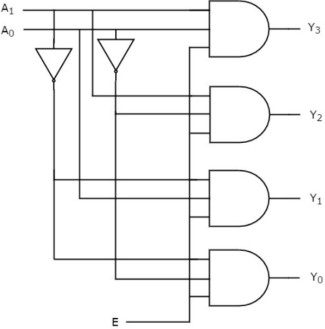


Figure 3: 1.1.1.2: 2 to 4 Decoder circuit

Also Encoder is an example of combinational circiut which performs the reverse operation of Decoder , so it passes 2N input lines and produce N output lines , the outputs are binary code equivalent to the input , we can represent the enable signal in encoder

This the block diagram of the Encoder

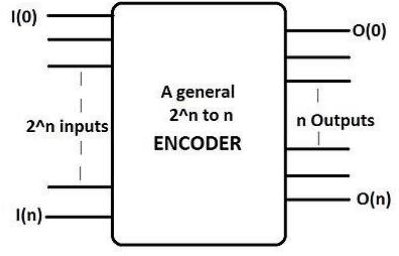


Figure 4: 1.2.1 : General encoder block diagram

Four to tow Encoder

Four to two encoder has four inputs Y3, Y2,Y1, and Y0 and two outputs A1 and A0, the figure is the block diagram of 4 to 2 Encoder

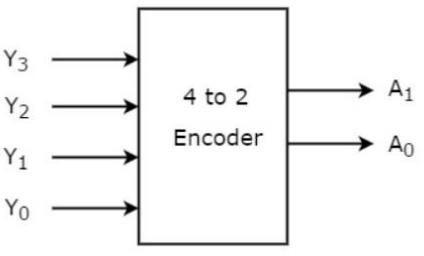


Figure 5: 1.2.1.1 : 4 to 2 Encoder

One of these inputs must be '1' to get a binary code at the output The truth table of 4 to 2 Encoder:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Outputs | | | | **Inputs** | |
| **Y3** | Y2 | Y1 | Y0 | A1 | A0 |
| **0** | 0 | 0 | 1 | 0 | 0 |
| **0** | 0 | 1 | 0 | 0 | 1 |
| **0** | 1 | 0 | 0 | 1 | 0 |
| **1** | 0 | 0 | 0 | 1 | 1 |

Table 2: 1.2.1.1: Truth table of 4 to 2 encoder

**Consrtuct 4 to 2 encoder using basic gated:**

We can implement 4 to 2 encoder using basic gates so we need two OR gates, this Figure is the circuit diagram of 4 to 2 encoder:

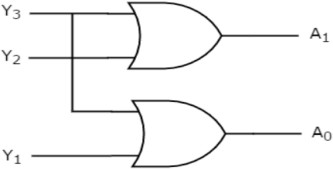


Figure 6: 1.2.1.2 : 4 to 2 Encoder circuit

In an encoder there are two problems , the first , there is no guarantee that only one of inputs is 1 and the second, if all inputs are 0 there is no case to this , so the priority Encoder solve these problems, the output of priority encoder depends on active input which has the highest priority , so when the highest priority input is present all other input with lower priority will be ignored.

Example of priority encoder: 8-input priority encoder, the following figure show 8 to 3 priority encoder block diagram with its truth table

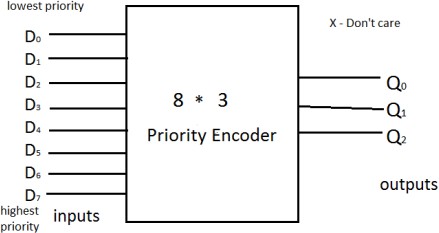


Figure 7: 1.2.2.1: 8 to 3 priority Encoder

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Digital inputs | | | | | | | | Binary outputs | | |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Q2 | Q1 | Q0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | x | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | x | x | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | x | x | x | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | x | x | x | x | 1 | 0 | 0 |
| 0 | 0 | 1 | x | x | x | x | x | 1 | 0 | 1 |
| 0 | 1 | x | x | x | x | x | x | 1 | 1 | 0 |
| 1 | x | x | x | x | x | x | x | 1 | 1 | 1 |

Table 3: 1.2.2.1: 8 to 3 priority Encoder truth table

## 1.3 Multiplexer:

A multiplexer is a combinational circuit ,it also known as Mux, Mux can have a maximum of 2n data inputs , and n selection lines and one output line , based on the binary value presented on the selection lines one of these inputs data transmit to the output , Since there are 'n' selection lines, the total number of possible combinations of zeros and ones is 2 n , so each combination will select only one data input and pass it to output.

Example of Mux is 4 to 1 that has four data inputs I3, I2, I1 and I0, two selection lines s1 and s0 and one output Y , the following figure is the 4x1 Mux block diagram:

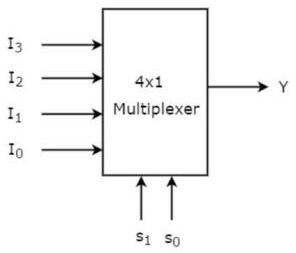


Figure 8: 1.3.1: 4 to 1 Mux block diagram

Only one of these 4 inputs will be passed to the output according to the combination of inputs present at these two selection lines, the following table is the truth table of 4x1 Mux

|  |  |  |
| --- | --- | --- |
| Selections | | Outputs |
| S1 | S2 | Y |
| 0 | 0 | I0 |
| 0 | 1 | I1 |
| 0 | 0 | I2 |
| 1 | 1 | I3 |

Table 4: 1.3.1: Truth table of 4 to 1 Multiplexer

**Construct 4x1 Mux using Basic gates:**

To implement 4x1 Mux we need inverters , AND gates and OR gates, the following figure show the circuit diagram of 4x1 Mux

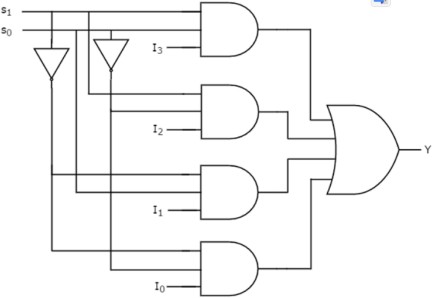


Figure 9: 3.1.2: 4 to 1 Multiplexer circuit

## 1.4 De-Multiplexer:

De-Multiplexer is a combinational circuit is also known as De-Mux , is the reverse operation of the Multiplexer , it has one input, n selection lines and maximum of 2n output , the input will be passed to one of these outputs according to the values of selection lines, as well as there are n selection lines then there will be 2 n combinations of zeros and ones, each combination passed to only one output

Example of De-Mux : 1x4 De-Mux , it has one input I , two selection lines s1 and s0 and four outputs Y3,Y2,Y1,Y0, the input I will be connected to one of these four outputs according to the values of selection lines s1 & s0, the following figure is show the block diagram of 1x4 De-Mux and its truth table

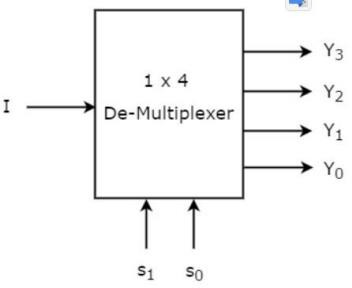


Figure 10: 1.4.1: 1 to 4 De-Mux block diagram

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Selections | | Outputs | | | |
| S1 | S0 | Y3 | Y2 | Y1 | Y0 |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Table 5: 1.4.1: Truth table of 1 to 4 De-Multiplexer

**Construct 1x4 De-Mux using Basic gates:** we can implement 1x4 De-Mux using basic gates so we need inverters and 3-input AND gates , the following figure show the circuit diagram of 1x4 De-Mux:

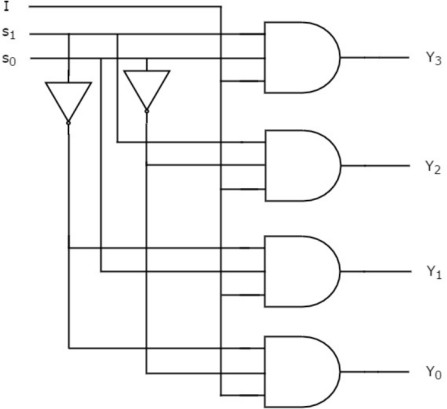


Figure 11 : 1.4.2: 1 to 4 De-Multiplexer

# 2.Procedure and Discussion

## 2.1 constructing a 4-to-2 Encoder With Basic Gates:

**2.1.1connection:**

We connect the circuit shown in the figure 2.1.1.1 and discussed the following points

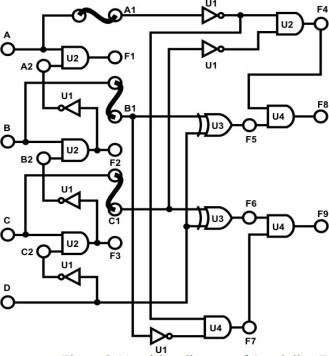


Figure 12: 2.1.1.1: wiring diagram of 4-to-2line Encoder

* + - * 1. We set the module IT-3004 and located block Encoder

2-We connected +5V of module IT-3004 to the +5V output of fixed power subbly section of IT-3000

3-We connected inputs A-D to Data switches SW0-SW3 recpectively , outputs F8 and F9 to Logic Indicator L0 and L1

4-We recorded the result in the following table 2.1.1.1

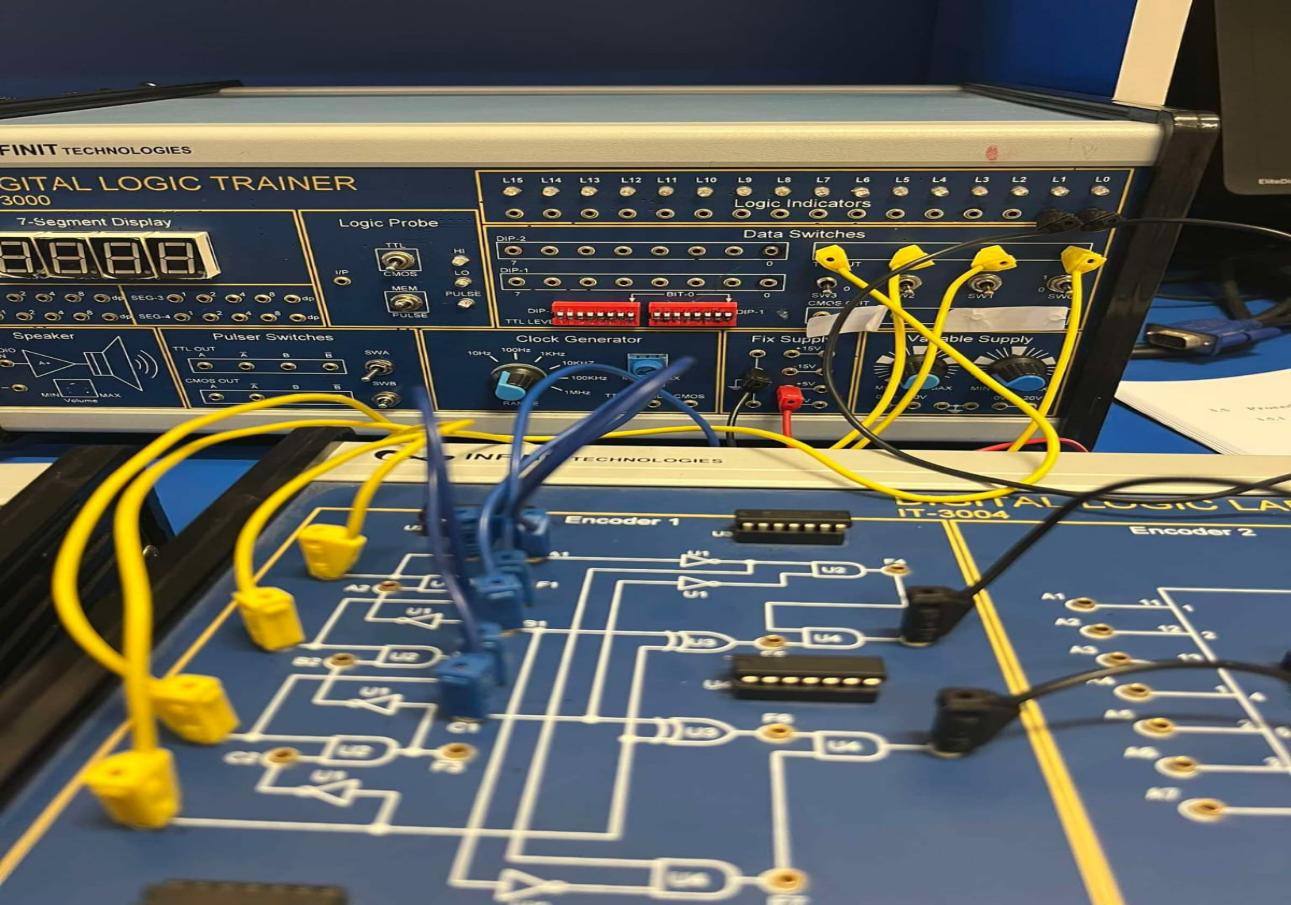


Figure 13: 2.1.1.2 : connection of 4-to-2 Encoder with basic gates

2.1.2 Results :

Table 6: 2.1.1.1: Data for 4-t-2 Encoder with basic gates

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Inputs | | | | Outputs | |
| D | C | B | A | F9 | F8 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

## **2.1.3**Discussion

As shown in the table, if more than one input is set to HIGH, the output is considered invalid. However, if only one input is set to HIGH, the output depends on this single HIGH input, producing a binary code equivalent to the active HIGH input.

## 2.2 Constructing 9-to-4-Line Encoder with TTLIC

### 2.2.1 connection

We connect the priority Encoder shown in the figure 2.2.1.1 and discussed the following points

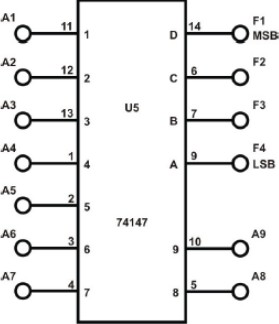


Figure 14: 2.2.1.1 : 74147 BCD Priority Encoder

1. We set the module IT-3004 and located 74147(U5) on block Encoder 2, and we connected +5V of module IT-3004 to the +5V output of fixed power supply
2. We connected inputs A1-A8 to DIP Switches 1.1-1.8 and A9 to SW0 , then we connected outputs F1-F4 to Logic indicators L1-L4
3. We recorded the result in the2.2.2.1 Table

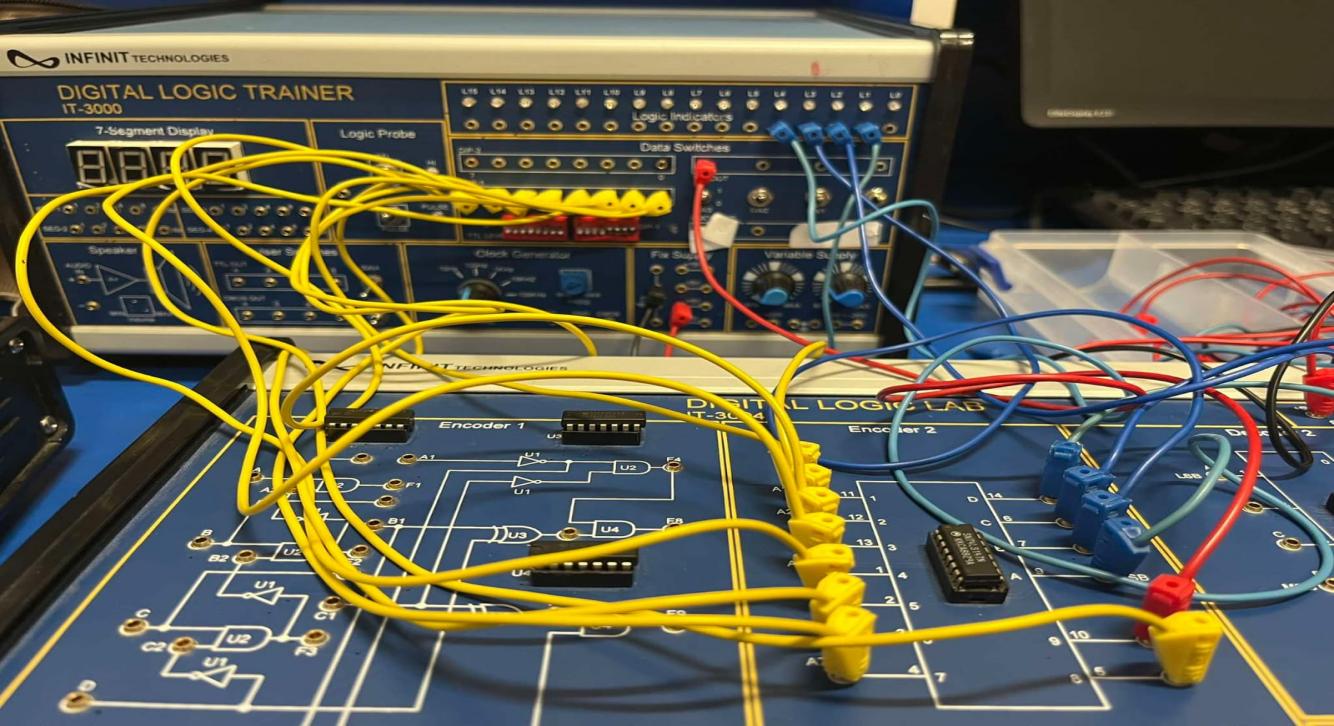


Figure 15: 2.2.1.2 : connection 9-to-4 Line Encoder

## 2.2.2 Results:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | | | | | | | Outputs | | | |
| A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | F4 | F3 | F2 | F1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Table 7: 2.2.2.1: the data of priority Encoder

## 2.2.3 Discussion :

Referring to the table, this is an active-low priority encoder, designed to address the limitations of a standard encoder. In case multiple inputs are active, the output is determined by the highest-priority active input, disregarding the others.

## 2.3Constructing 2-to-4 Line Decoder with basic gates

**2.3.1Connection:**

We connected following circuit in Figure 2.3.1.1

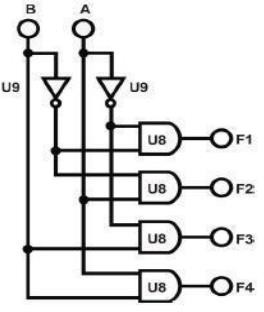


Figure 16: 2.3.1.1: 2-t-4 decoder

1-Block Decoder 1 of module IT-3004 will be used in this section of the experiment.

Connect +5V of module IT-3004 to the +5V output of fixed power supply

2-We Connected inputs A and B to Data Switches SW0 and SW1. Connect outputs F1~F4 to Logic Indicators L0~L3

3-We recorded the result in 2.3.1.1Table

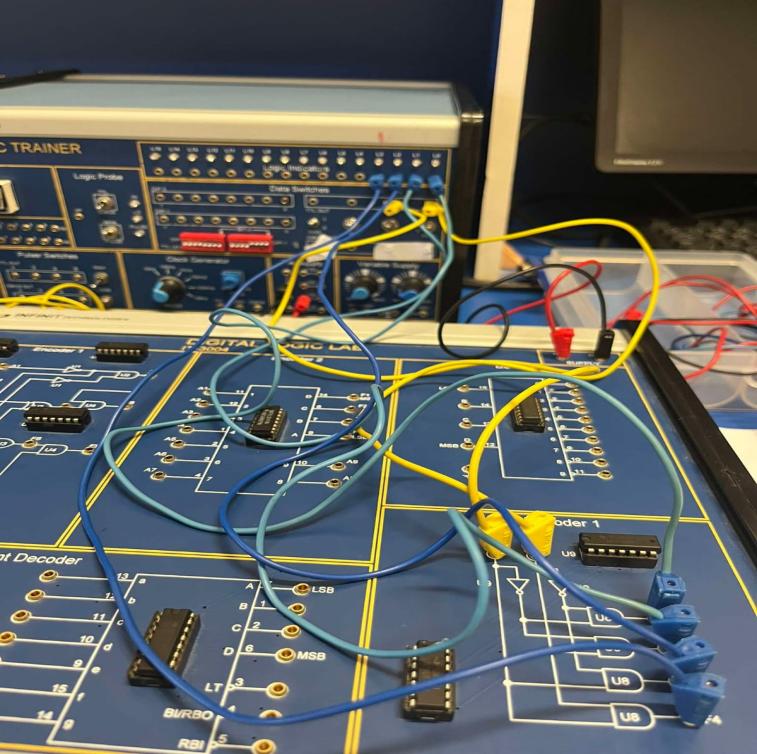


Figure 17: 2.3.1.2: connection 2-to-4 Line Decoder

2.3.2Results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Inputs | | Outputs | | | |
| B | A | F1 | F2 | F3 | F4 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

Table 8: 2.3.2.1: 2-to-4 line Decoder with basic gates

## 2.3.3Discussion:

Examining the table results, we observe that for each input combination, only one output is set to HIGH while the rest remain LOW. This corresponds to the specific combination number. For instance, when the number is 0, F1 is set to HIGH and the rest are LOW. Similarly, if the number is 1, F2 is HIGH and the rest are LOW. This pattern continues where each number corresponds to a specific output being HIGH while the others are LOW.

## 2.4Constructing 4-to-10 Line Decoder with TTLIC:

**2.4.1Connection:**

We connected the following block circuit in figure 2.4.1.1

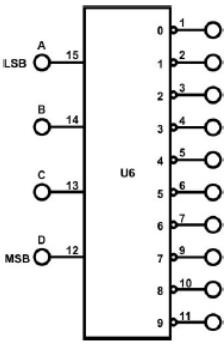


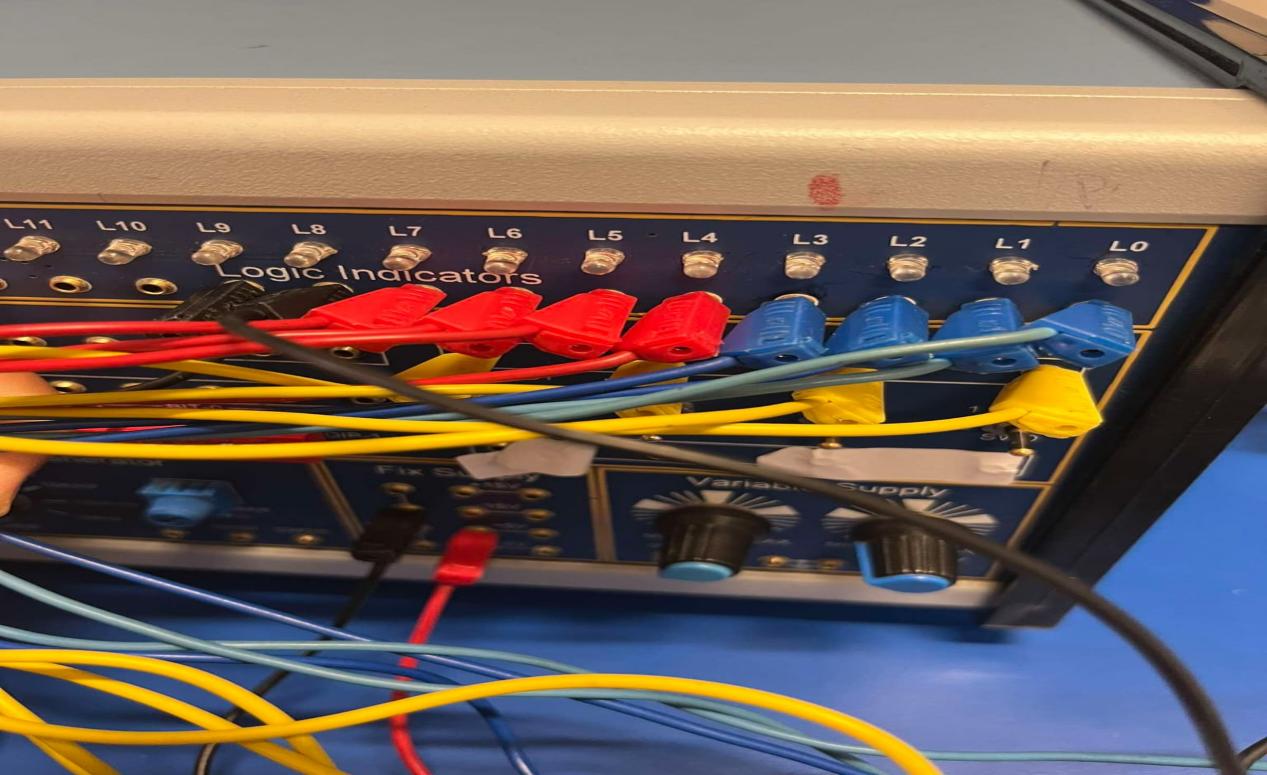
Figure 18: 2.4.1.1: 4-to-10 Line Decoder

1-We set the module IT-3004 and located U6(7442) on block decoder 2

2- We Connected inputs A-D to the Data Switches SW0-SW3, respectively and Connected

10 outputs to corresponding Indicators L0-L9

3- We recorded the results in the 2.4.2.1 table



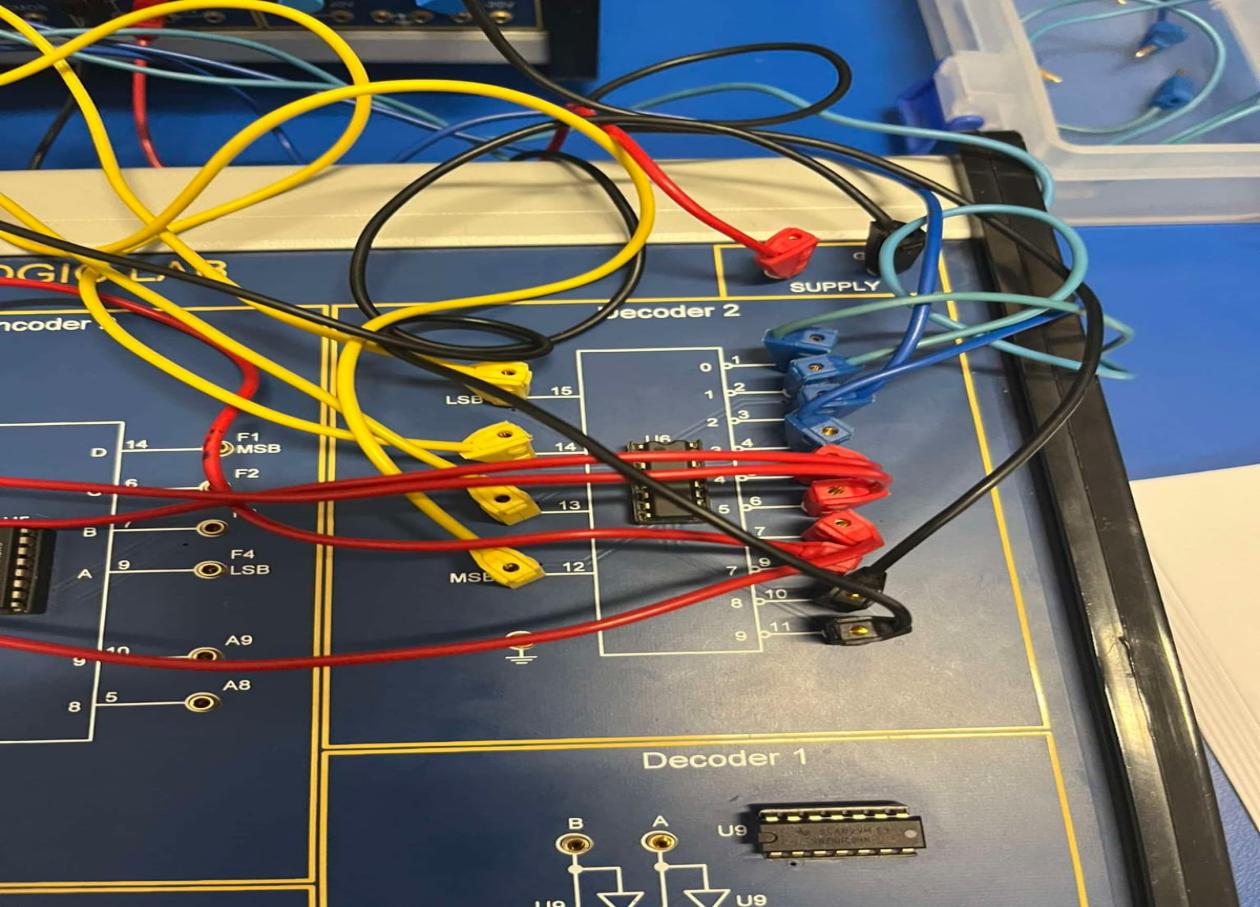


Figure 19: 2.4.1.2 : conection 4-to-10 Line Decoder

## 2.4.2Results:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Inputs | | | | Outputs | | | | | | | | | |
|  | D | C | B | A | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Table 9: 2.4.2.1 : Data for 4-to-10 Line Decoder

## 2.4.3Discussion:

Looking at the table, it's clear this is an active LOW decoder. Here, only one output is set to LOW while the rest are HIGH, depending on the input combination number. For instance, if the combination number is 0, L0 is LOW and the others are HIGH. If it's 1, then L1 is LOW and the others are HIGH, and so on. This pattern continues, with each combination number corresponding to a specific output being LOW while the others are HIGH,

## 2.5Constructing 2-to-1 Line Multiplexer with Basic Gates:

**2.5.1Connection:**

We connected the following circuit in the figure 2.51.1

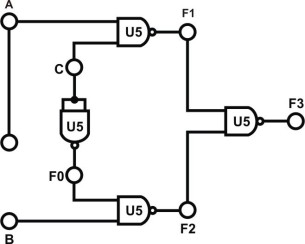


Figure 20 : 2.5.1.1: 2-to-1 Multiplexer

1-We used block Multiplexer 1 of module IT-3005 as a 2-to-1 MUX and we Connected +5V of module IT-3005 to the +5V output of fixed power supply

2 -We recorded the result in 2.5.2.1 table

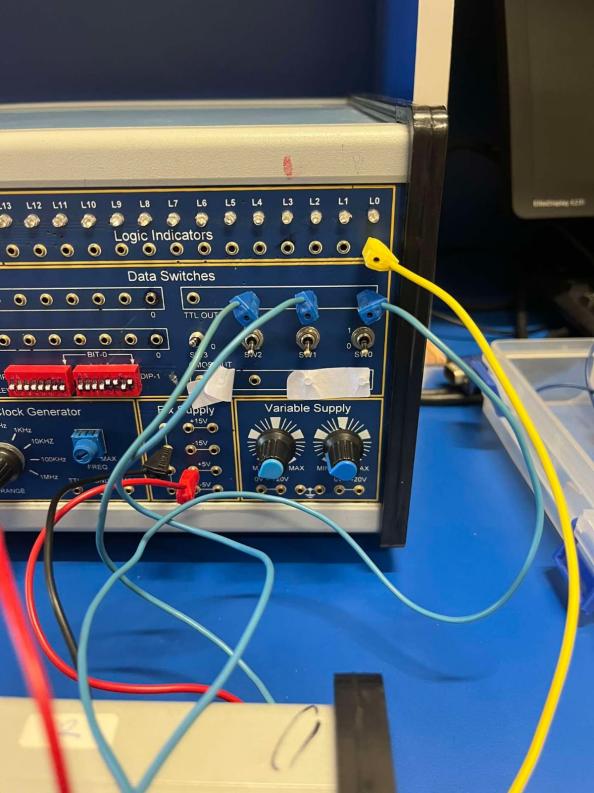
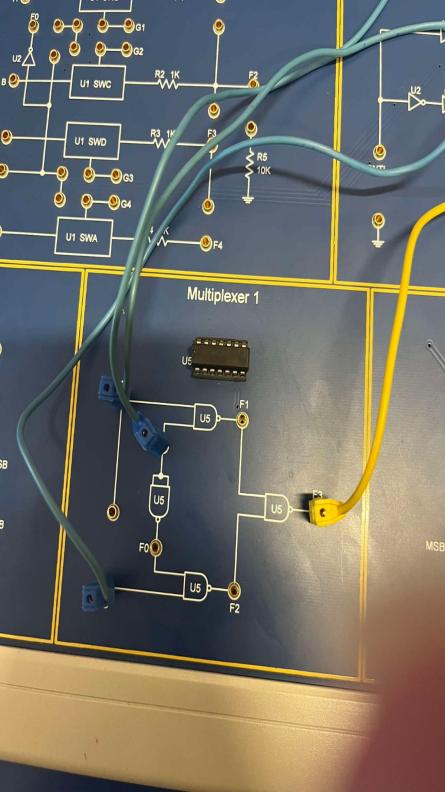


Figure 21: 2.5.1.2 : connection 2-to-1 Line Multiplexer with basic gates

## 2.5.2Results:

|  |  |  |  |
| --- | --- | --- | --- |
| Inputs | | | Outputs |
| C | A | B | F3 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Table 10: 2.5.2.1 : 2-to-1 Line Multiplexer with Basic Gates

## 2.5.3Discussion:

From the table, we observe that the output of the Mux corresponds to one of its inputs. In simpler terms, the selected input is connected to the output, and this selection is determined by the setting of the selection lines. In this multiplexer, the selection line is labeled as C, while the inputs are denoted as A and B. If C is LOW, then input B is connected to the output F3. Conversely, if C is HIGH, then input A is connected to the output

## 2.6Constructing 8-to-1 Line Multiplexer with IC

2.6.1Connection

We connected the following circuit diagram in the figure 2.6.1.1

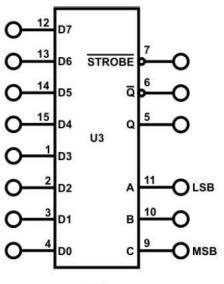


Figure 22: 2.6.1.1 : 8-to-1 Mux

1-We used U3 (74LS151) on block Multiplexer 2 of module IT-3005

2-We connected inputs D0-D7 to DIP Switch 1.0-1.7; inputs C, B, A to Data Switches SW2, SW1, SW0

3-We recorded results in 2.6.2.1 table

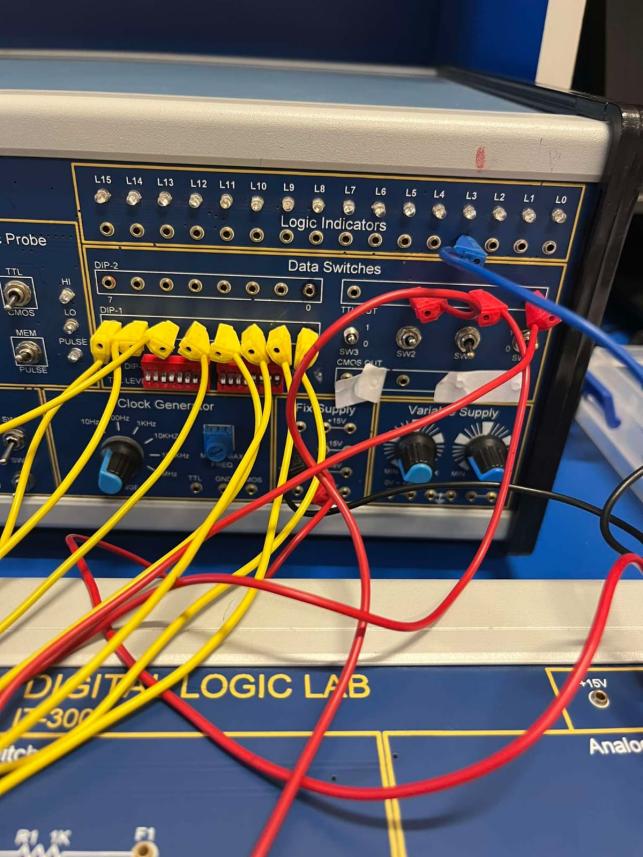
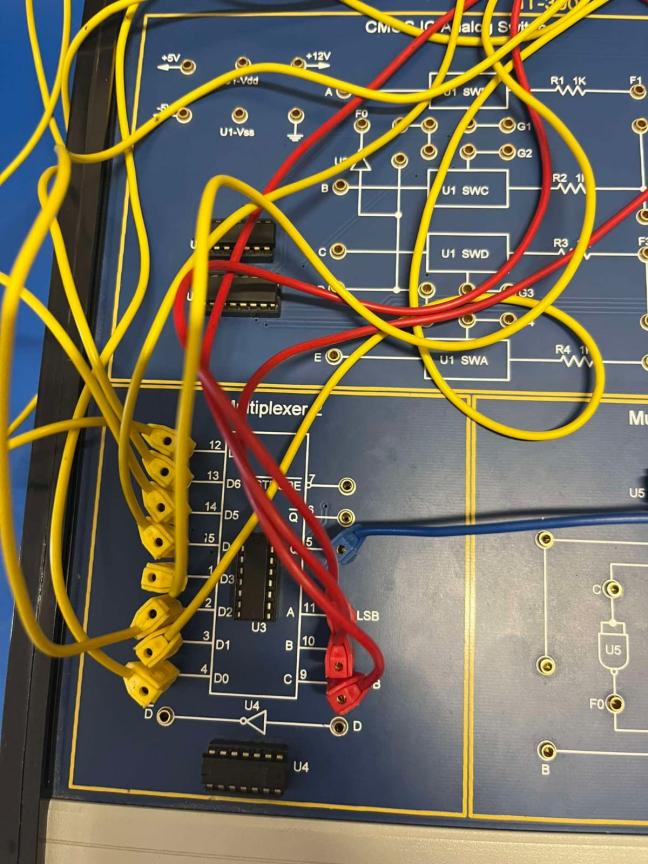


Figure 23: 2.6.1.1 : connection 8-to-1 Line Multiplexer with IC

## 2.6.2 Results

|  |  |  |  |
| --- | --- | --- | --- |
| Inputs | | | outputs |
| C | A | B | Q |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | I |
| 0 | 1 | 1 | I |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Table 11: 2.6.2.1 : Data for 8-to-1 Line Multiplexer

## 2.6.3 Discussion :

31As shown in the table, this once again confirms that only one of the Mux's inputs is connected to the output based on the settings of the selection lines. In this Mux, the selection lines are labeled as C, B, and A, while the inputs are labeled as I0 through I7. One of the inputs, ranging from I0 to I7, is connected to the output Q

## 2.7Constructing 1-to-2 Line Demultiplexer with Basic Logic Gates:

**2.7.1Connections**:

We connected the following circuit in the 2.7.1.1 figure

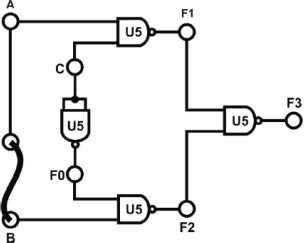


Figure 24: 3.17 : 1-to-2 Demultiplexer

1- We used Block Multiplexer 1 of module IT-3005 , we connected A to Data Switch SW0, C to SW3; F1 and F2 to Logic Indicators L1 and L2 respectively

2- We set the value of C to 0 and change the value at input A , and then set the value of C to 1 and change the value at input A ,finally we recorded the results in 2.7.2.1 Table

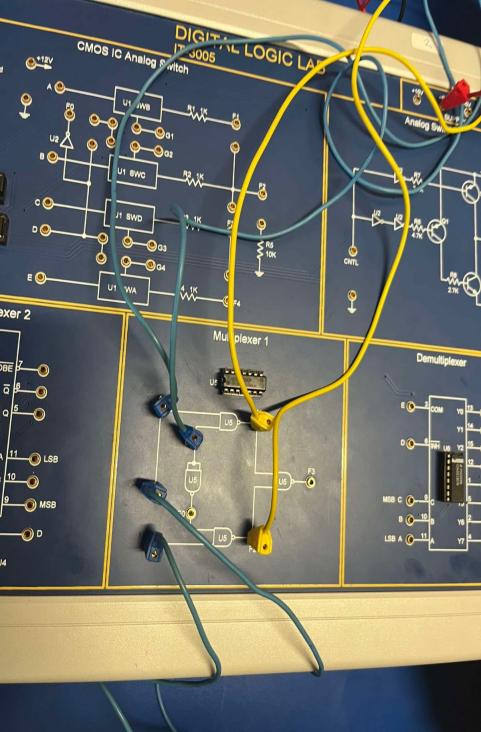
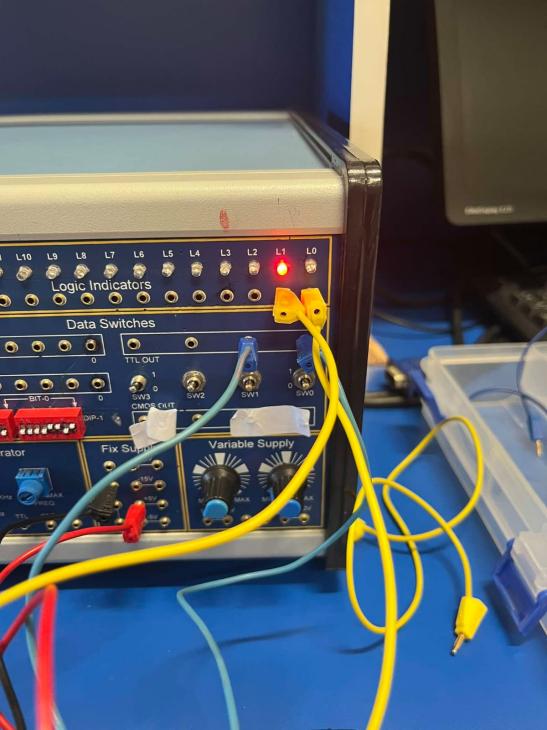


Figure 25: 2.7.1.2 : connection 1-to-2 Line Demultiplexer with basic gates

## 2.7.2Results:

|  |  |  |  |
| --- | --- | --- | --- |
| Inputs | | Outputs | |
| C | A | F1 | F2 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |

Table 12: 2.7.2.1 : Data for 1-to-2 Line DeMux

## 2.7.3Discussion:

As the result shown in table , this is an active LOW DeMux , C is selection and A is input , so when C is 0 then F2 is the complement value of A , and if C is 1 then F1 is the complement value of A

## 2.8Constructing 1-to-8-Line Demultiplexer with CMOS IC

### **2.8.1 Connection:**

We connected the block circuit in the following 2.8.1.1Figure

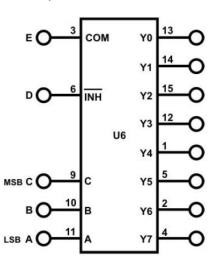


Figure 26: 2.8.1.1: 1-to-8 Demultiplexer

1-We used U6 (4051) on block Demultiplexer of module IT-3005 Connect +5V, -5V of module IT-3005 to the +5V and -5V output of fixed power supply respectively.

2-We connected E to DIP 1.0, D to DIP1.1 , A to SW0 , B to SW1 , C to SW2 , outputs Y0-Y7 to Logic Indicators L0-L7 respectively

3-We changed the values of D and E and recorded the result in 2.8.2.1 table

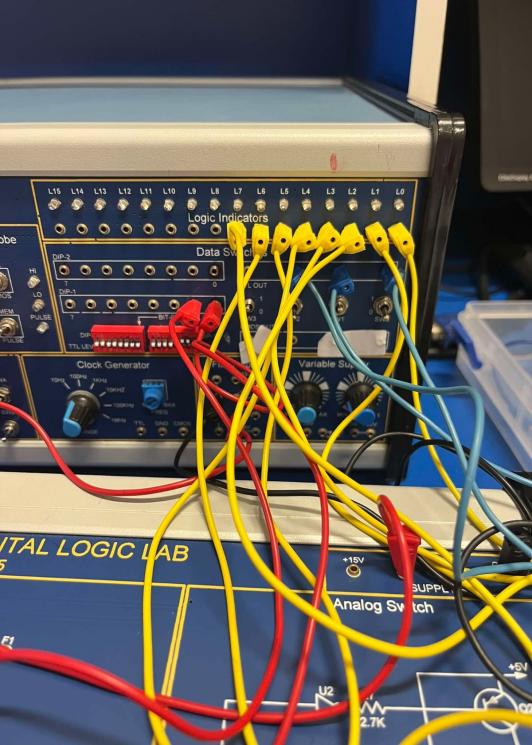
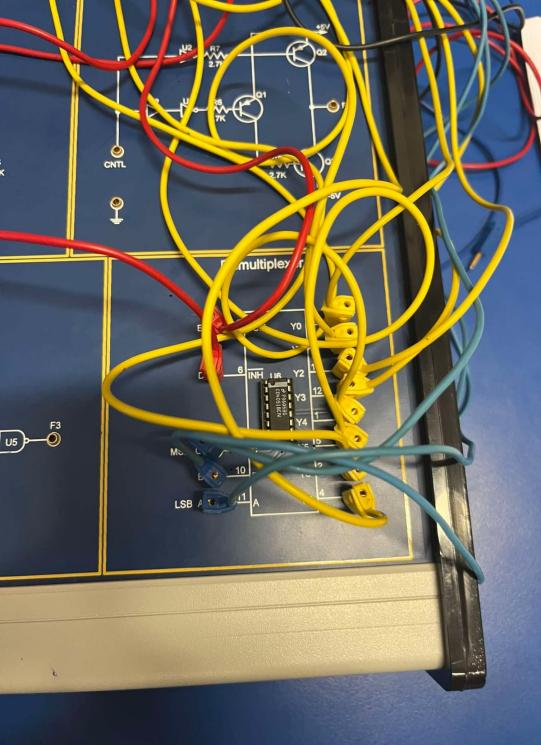


Figure 27:2.8.1.2: connection 1-to-8-Line Demultiplexer with CMOS IC

2.8.2 Results

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | Outputs | | | | | | | |
| C | B | A | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 13: 2.8.2.1 : data for 1-to-8 DeMux

## 2.8.3 Discussion :

As indicated by the results in the table, C, B, and A serve as the selection lines, while Y0 -Y7 represent the outputs. E acts as the input, and D is an active LOW Enable (meaning the circuit operates when D is LOW and does not work when HIGH). Therefore, when D is LOW, only one of the outputs is connected to the input E, determined by the binary code present on the selection lines. For instance, when the binary code is 000, Y0 is equal to E, and the remaining outputs are set to zero, and so on

# Conclusion:

After completing this experiment, all objectives were successfully achieved, enabling the construction of combinational circuits such as Decoders, Encoders, Multiplexers, and Demultiplexers using basic gates. To implement a Decoder, we used several AND gates and INVERTER gates. An Encoder required a few OR gates. A Multiplexer involved a single OR gate, several AND gates, and a few INVERTER gates. For a Demultiplexer, multiple AND gates and INVERTER gates were necessary.

Additionally, we learned to build these circuits using block modules. This method simplifies the process as you only need to connect the inputs and outputs, with the complex circuitry already integrated within each block. This approach made constructing these circuits much easier and faster.

As a result, I have made significant progress in the practical aspect of constructing circuits.

# References:

1. : Manual for Digital Electronics and Computer Organization Lab, 2023-2024, Birzeit University
2. : UTMEL

[Accessed on 17/7/2024] https://[www.utmel.com/blog/categories/integrated%20circuit/what-is-a-decoder](http://www.utmel.com/blog/categories/integrated%20circuit/what-is-a-decoder)