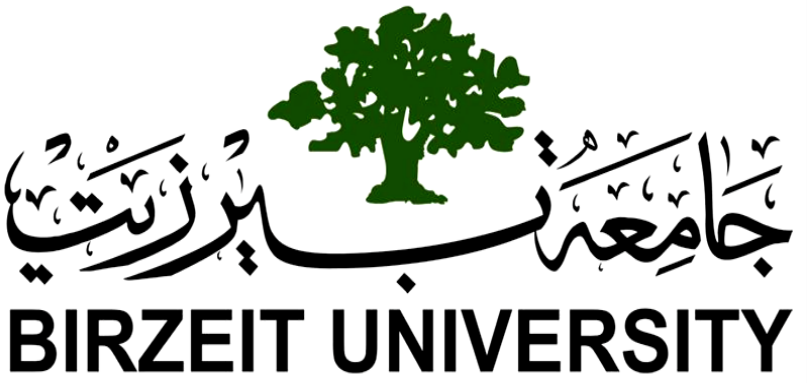


**Faculty of Engineering & Technology**
**Electrical & Computer Engineering Department**

# DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY - ENCS2110

## Report #2

## Encoders, Decoders, Multiplexers and Demultiplexers

---

**Prepared by:**

**Obada Tahayna** **1191319**

**Partners:**

**Abdallah Bahrawi** **1172316**

**Nour Tuffaha** **1192106**

**Instructor**: Dr. Adnan Yahya
**Assistant**: Eng. Katy Sadi

**Section:** 5
**Date:** 28/09/2020

# Abstract

In this experiment we will learn and discuss about some of the circuit design parts which included (Encoders, Decoders, Multiplexers and Demultiplexers), so we will see how we could implement it, how it works, what the design of each one, the important of each one and why use it in circuit. Our aim in this experiment is: • To understand the operating principles of Encoders/Decoders.

• To understand the operating principles of Multiplexers/Demultiplexers.

• To construct encoders and decoders using basic gates and IC.

• To construct multiplexers and demultiplexers using basic gates and IC.

During this experiment we found that every design has a different work from others and we could build any design from basic logic gates or from anther design with some basic logic

# Contents

# Theory

We have 2 aims we will depend on it during this experiment and will see every one alone:

## 1-To understand the operating principles of Encoders/Decoders.

So, in the first aim we will discuss Decoder and Encoder: Decoder is a combinational circuit that has ‘n’ input lines and maximum of $2^n$ output lines. One of these outputs will be active High based on the combination of inputs present, when the decoder is enabled. That means decoder detects a particular code. The outputs of the decoder are nothing but the min terms of ‘n’ input variables (lines), when it is enabled.[1]

The decoder work as Code translator, which translates a binary-coded decimal value into the corresponding segment control signals for input integer values 0 to 9.

There are many types of decoders which depends on input ‘n’, 2 to 4 Decoder is the most famous one of decoder which take 2 input and give 4 output and Enable (E) as extra input see figure 1.

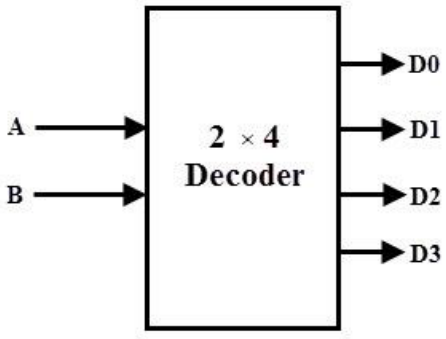


*Figure 2: 2 to 4 Decoder without E.*

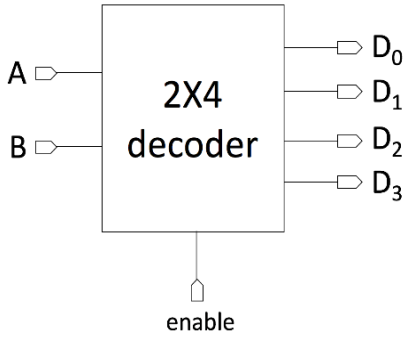


*Figure 1: 2 to 4 Decoder with E.*

An Encoder is a combinational circuit that performs the reverse operation of Decoder. It has maximum of $2^n$ input lines and ‘n’ output lines. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2n input lines with ‘n’ bits. It is optional to represent the enable signal in encoders.[2]

In simple the Encoder is the opposite of Decoder.

The Encoder work to converts information from one format or code to another, for the purpose of standardization, speed or compression.

As like Decoder the 4 to 2 Encoder is famous, which takes 4 input and return 2 output see figure 3.

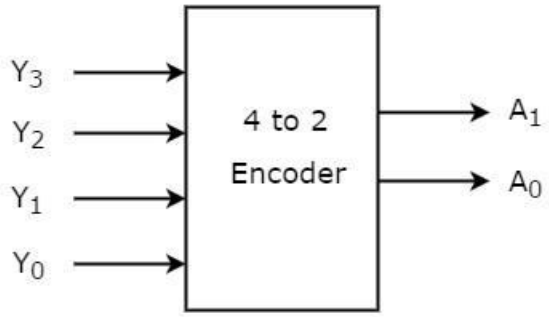


*Figure 3: 4 to 2 Encoder*

## 2- To understand the operating principles of Multiplexers.

Multiplexer (or mux) is a device that combines several analog or digital input signals and forwards them into a single output line. A multiplexer of inputs has selected lines, which are used to select which input line to send to the output. Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth. Multiplexers can also be used to implement Boolean functions of multiple variables.[3]

Mux 2 to 1 is the most famous see figure 4:

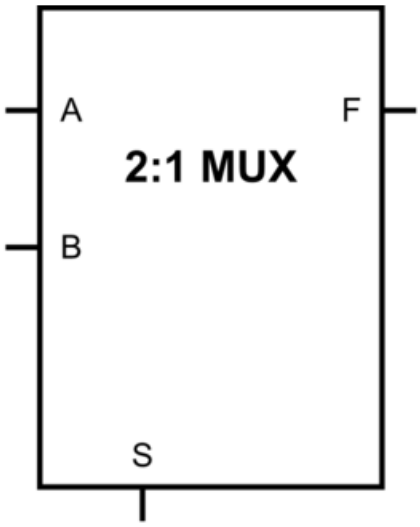


*Figure 4: 2*1 Multiplexer.*

# Procedure

## A. Constructing4-to-2-LineEncoderwithBasicGates

First, we will connect IT-3000 Electric Circuit with IT-3004 Circuit by wire in +5V power supply for both in IT-3004 there is a part which write on it Encoder and have a basic logic gates that present the Encoder Circuit as Figure 5.

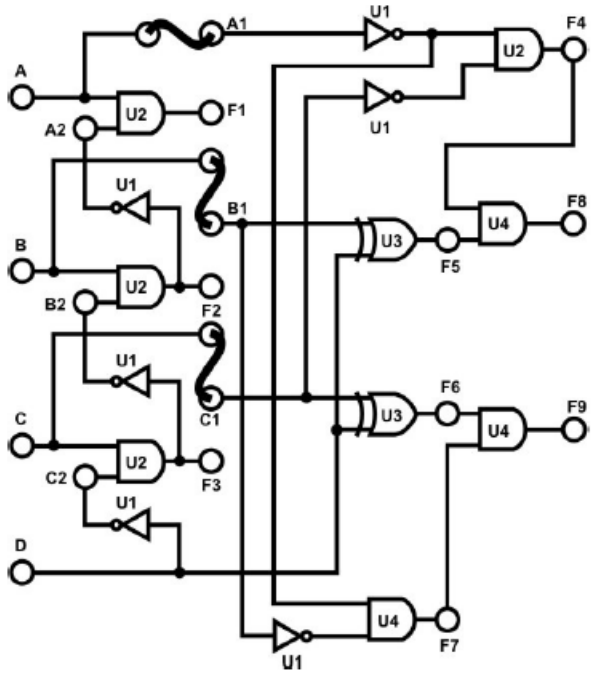


*Figure 5: wiringdiagramof4-to-2lineEncoder*

so, this is the basic logic gates that present encoder as we see in figure 5 form (A-D) input and (F4, F5, F6, F7, F8, F9) output but we will use F8, F9 as output, now we connect the input from A to D with Switches in IT-3000 and F8, F9 with Logic Indictor L1, L2.

We have 4 input and 2 output this mean we have 16 status we see and try as show in the truth table 1:

*Table 1: 4-to-2 Truth Table*

| 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 |

We note that this circuit works as 4*2 encoder only when one of input=1 and the others status the output=0, But we must attention the input =1 depends in more significant.

Input (DCBA): output (L2, L1)

0001: 00.

0010: 01.

0100: 10.

1000: 11.

## B. Constructing 9-to-4-LineEncoderwithTTLIC

After we did the previous circuit, we need to see the encoder but this time with more input and output so we will be connected 9 input and we have 4 outputs, as in figure 6 we will connect the new circuit on IT-3004 U5 74147 and see the table for it.

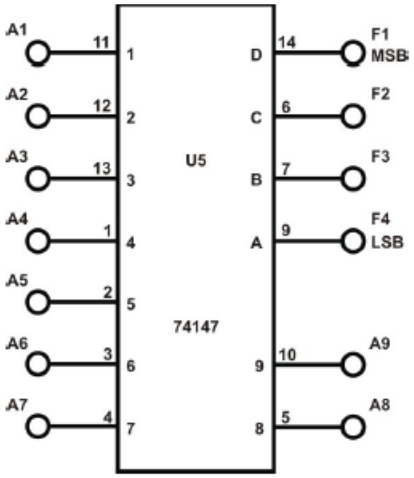


*Figure 6: (74147)BCDPriorityEncoder*

This figure show that we will connect the input from A1-A8 to the switches DIP switches 1.0-1.7 (important) and the last input A9 to 2.0. The output from F1 to F4 with L1 to L4 the table 2 see the output we have when try this circuit.

*Table 2: 9-to-4-LineEncoderwithTTLIC Truth Table*

| 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 | 0 | **1** | **1** | **1** |
| 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 | 0 | 1 | 1 | 1 | **1** | **1** | **1** | 0 |

## C.Constructing2-to-4 LineDecoderwithBasicGates

In the same IT-3004 we will see Decoder built from Basic Gates and this what we need to circuit. Power supply on +5V, we have two input (A, B) connected in Switches W0, W1, and we have 4 output (F1, F2, F3, F4) Connected with L1, L2, L3, L4.

Figure 7 will see the basic gates that Decoder components and Table 3 the output we have after connected.

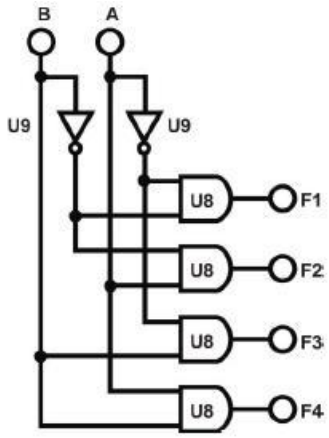


*Figure 7: 2-to-4Decoder*

*Table 3: 2-to-4Decoder Truth Table*

| 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** |

## D. Constructing 4-to-10 LineDecoderwithTTLIC

In the same IT-3004 block Decoder 2 U6, will be using in this section of a new circuit but not a binary number, it using BCD-to –Decimal decoder IC.

We have 4 input (A, B, C, D) and 10 output (0 to 9), we connected input A-D with Switches W0-W3 and the output with L0-L9.

Figure 8 show the BCD decoder will use, and Table4 give us the result doing this section of experiment.

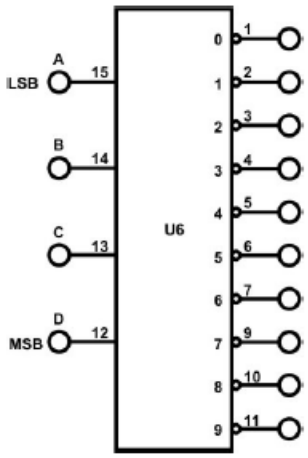


*Figure 8: 4-to-10Decoder*

*Table 4: -to-10Decoder Truth Table*

| | Input | | | | Output | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 |

## E. Constructing2-to-1-LineMultiplexerwithbasicGates

In this part of experiment, we will move to next block (module) IT-3005, this module used for Multiplexer and Demultiplexer, first of all we will started with Multiplexer with basic gates component.

Bring IT-3005 and go on it like in figure 9 that contains mux with basic gates, as we see in figure 9 we have 2 input (A, B) and 4 output (F0, F1, F2, F3) and we use F3 as output now and we have another input selected (C), as we do previous connected A,B with Switches W0, W1, selected in W2 and F3 with L1 we have 8 status depends on C if it 1 or 0, in table 5 we see the result.

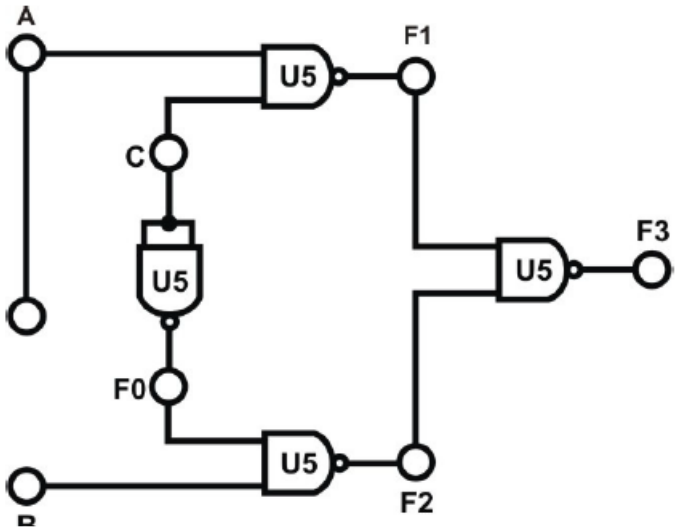


*Figure 9: 2-to-1MUX*

*Table 5: 2-to-1Mux Truth Table*

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

## F. Constructing 8-to-1 LineMultiplexerwithIC

Now in the same module IT-3305, a new part U3 (47LS151), in this part we have 8input (D0-D7) and another 3 input BCD (A, B, C) and one output Q. During this part when:

CBA= 000, data at D0 is send to output Q.

CBA =001, data at D1 is sent to output Q. . . .

CBA= 111, data at D7 is sent to output Q.

Note that this IC have function properly only when STROBE='0', when STROBE='1' IC don't change output according to input, Q still = '1'.

Now we connect inputs (D0-D7) with Switch 1.0-1.7, (ABC) with Switches W0, W1, W2 and output Q with L1.

As we see in Figure 10 and Table 6 how we could connect and the result of this part Result it is which of input D0-D7 does Q depend on.

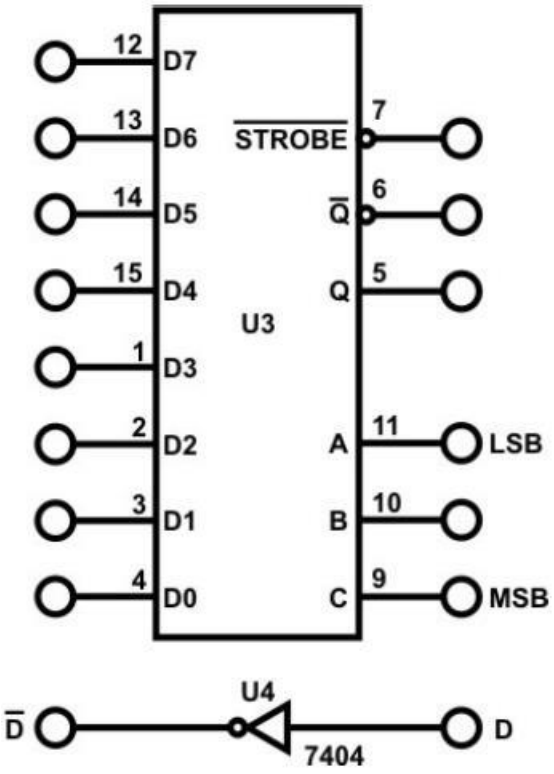


*Figure 10: 8-to-1 MUX*

*Table 6: 8-to-1 MUX Truth Table*

| C | B | A | Q |
|---|---|---|---|
| 0 | 0 | 0 | D0 |
| 0 | 0 | 1 | D1 |
| 0 | 1 | 0 | D2 |
| 0 | 1 | 1 | D3 |
| 1 | 0 | 0 | D4 |
| 1 | 0 | 1 | D5 |
| 1 | 1 | 0 | D6 |
| 1 | 1 | 1 | D7 |

## G. UsingMultiplexertoCreate a Logic Function

Exercise: The U3 (47LS151)) will used to create the function:

F (A, B, C, D) =Σ (0,2,4,5,7,8,10,11,15) 3. (Figure.10) the input connections necessary to implement the function F (A, B, C, D) =Σ(0,2,4,5,7,8,10,11,15)We Completed the connection by referring to the wiring diagram in Figure10, and we connect inputs (A,B,C,D) to Data Switches W0,W1,W2,W3 respectively. Output Y was connected to Logic Indicator L1.

As we see the implement in table 7:

*Table 7*

| A | B | C | D | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **1** |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | **1** |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | **1** |
| 0 | 1 | 0 | 1 | **1** |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | **1** |
| 1 | 0 | 0 | 0 | **1** |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | **1** |
| 1 | 0 | 1 | 1 | **1** |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | **1** |

We consider A, B, C to be the selection lines and D for the input We can note from the previous table 7 that there is a relation between D and the output, we have 8 combinations, (D', D', 1, D, D', 1, 0, D).

## H. Constructing1-to-2LineDemultiplexerwithBasicLogicGates

For this section from experiment we will using basic logic of MUX 1 of module IT-3005 to make Demultiplexer. As in figure 11 the Demultiplexer has the same basic logic gates of mux but the different here we connected the input B with wire in input A.

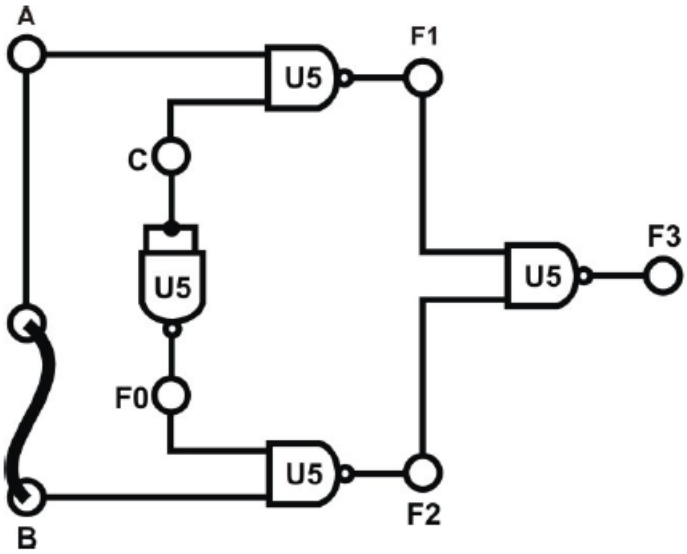


*Figure 11: 2*1Demultiplexer.*

we connected the inputs (A, B, C) with Switches W0, W1, W2 and the output F1, F2 with L1, L2 and connected also B as in figure we don't connected F3 as output. Now we while set C=0 and change A, and set C=1 and change A and see the how F1, F2 change with changing A in the two case, see Table 8 to know the change that happened.

| C | B | A | F1 | F2 |
|---|---|---|---|---|
| 0 | x | 0 | **1** | **1** |
| 0 | x | 1 | **1** | 0 |
| 1 | x | 0 | 0 | **1** |
| 1 | x | 1 | **1** | **1** |

## I. Constructing1-to-8-LineDemultiplexerwithCMOSIC

In the last part of experiment, we used U6(4051) on block Demux of module IT-3005 as see in figure 12 we have 1 input ,2 selection line and 8 output, notes that we must connect +5V, -5Vof module IT-3005 with +5V, -5V of power supply.

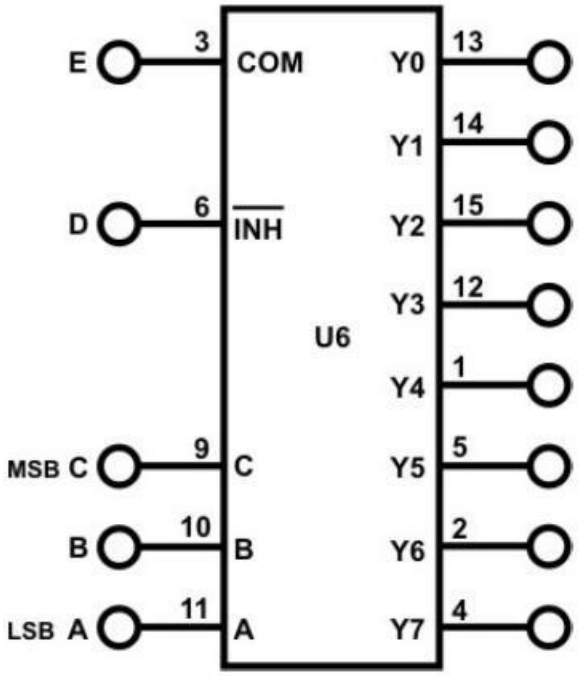


*Figure 12: 1-to-8DEMUX*

Now we see in figures some input (E, D, C, A, B) so we connected these inputs as follow: E to DIP 1.0 --- D to 1.1 --- A to W0 --- B to W1 --- C to W2, Output (Y0-Y1) to L1-L8. Now we put D=0 and D=1 and see if the output change as input. When D=0 the Table 9 give us the change that happened when change the sequence of input (A, B, C).

| 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** |

## Conclusion

At the end, we understood the basic concepts of Decoders, Encoders, Multiplexers and Demultiplexers. We also knew how to build logic functions using decoders and multiplexers. We proved and analyzed the operating techniques of Decoders, Encoders, Multiplexers and Demultiplexers.

# References

[1] https://www.tutorialspoint.com/digital_circuits/digital_circuits_decoders.htm. Accessed on 4/10/2020 at 07:12PM.

[2] https://www.tutorialspoint.com/digital_circuits/digital_circuits_decoders.htm. Accessed on 4/10/2020 at 09:01PM.

[3] https://en.wikipedia.org/wiki/Multiplexer. Accessed on 05/10/2020 at 02:55AM.

[4] ENCS211 Lab Manual. Accessed on 5/10/2020 at 06:32PM.