ENCS2110

-Department of Electrical and Computer Engineering.

-Digital electronics and computer organization laboratory.

- Report: Experiment No. 3 - Encoders, decoders, multiplexers, and demultiplexers logic circuits.

- Student name: Abdulrahman Shaheen.

- ID#: 1211753.

- Instructor Name: Dr. Qadri Mayyala

- Date: 19/4/2023.

- Section: 2.

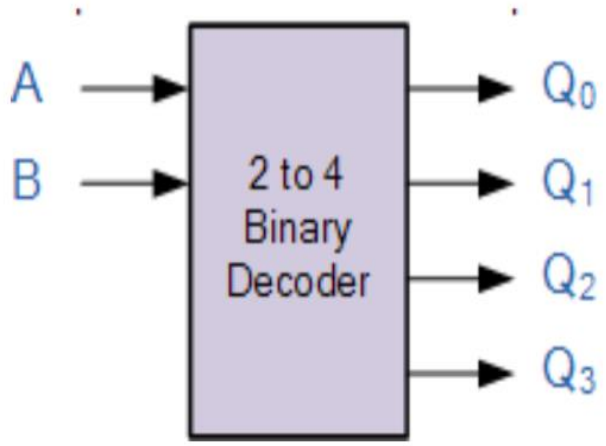
Abstract:

The goal of this experiment is to understand encoders, priority encoders, decoders, multiplexers and demultiplexers. And how to build these devices using basic gates. Also using IC.

Brief theoretical review:

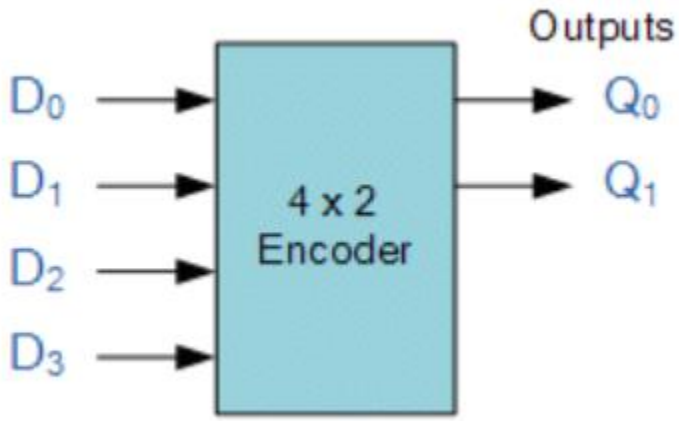
*Decoder:

A decoder is a circuit that changes a code into a set of signals. which takes an n-digit binary number and decodes it into 2^n data lines.



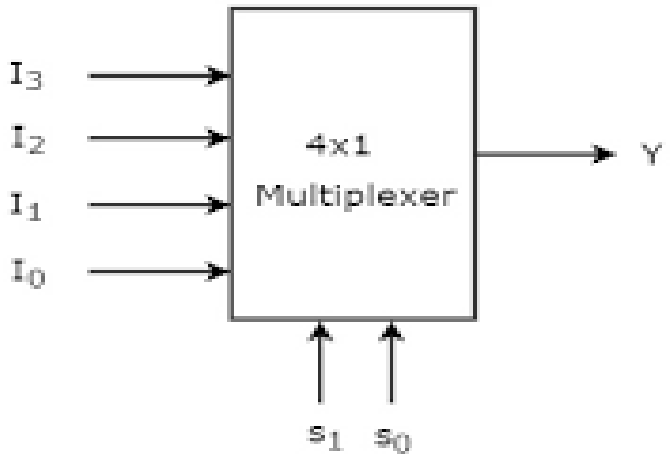
4

*Encoder: An encoder is a digital circuit that converts a set of binary inputs into a unique binary code. The binary code represents the position of the input and is used to identify the specific input that is active.



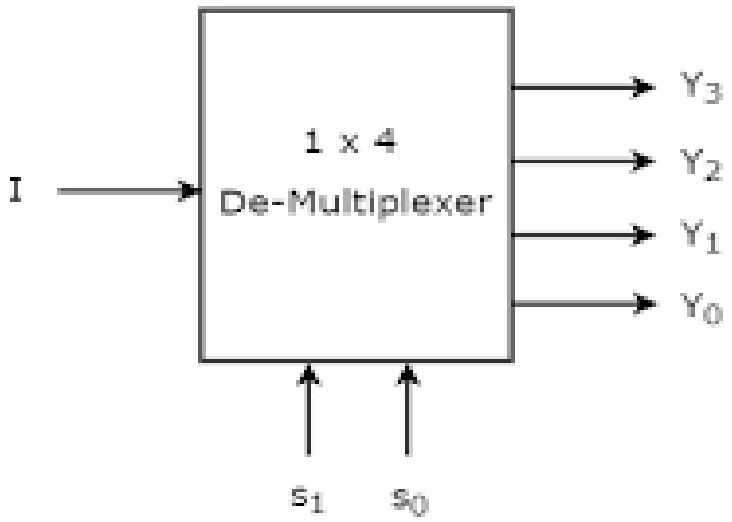
* Multiplexer:

a Multiplexer, also known as a data selector, is a device that selects between several analog or digital input signals and forwards the selected input to a single output line



* De-Multiplexer:

is a circuit that has one input and more than one output. This description sounds like the description given for a decoder, but a decoder is used to select among many devices while a demultiplexer is used to send a signal among many devices.



*Prelab:

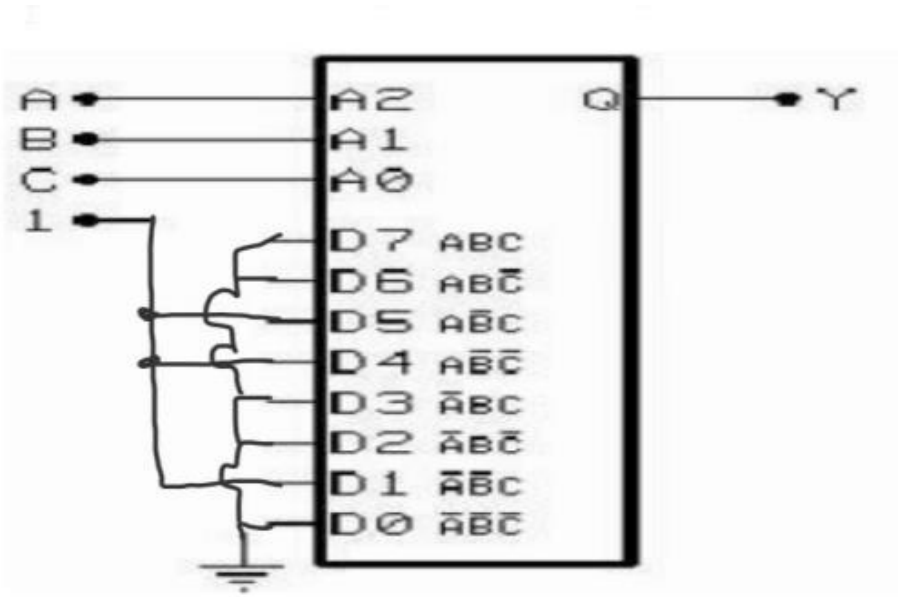
Q1) a) Design a circuit which uses an SN74151 to implement a sum-of products expression, as follows: Convert the following expression into summation form (i.e., F (A, B, C) =∑ (…)): y = f (A, B, C) = AB’ + B’C.

F=AB’C+AB’C’+AB’C+A’B’C F= ∑ (1,4,5).

F= ∑ (1,4,5).

Inputs			Outputs
A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

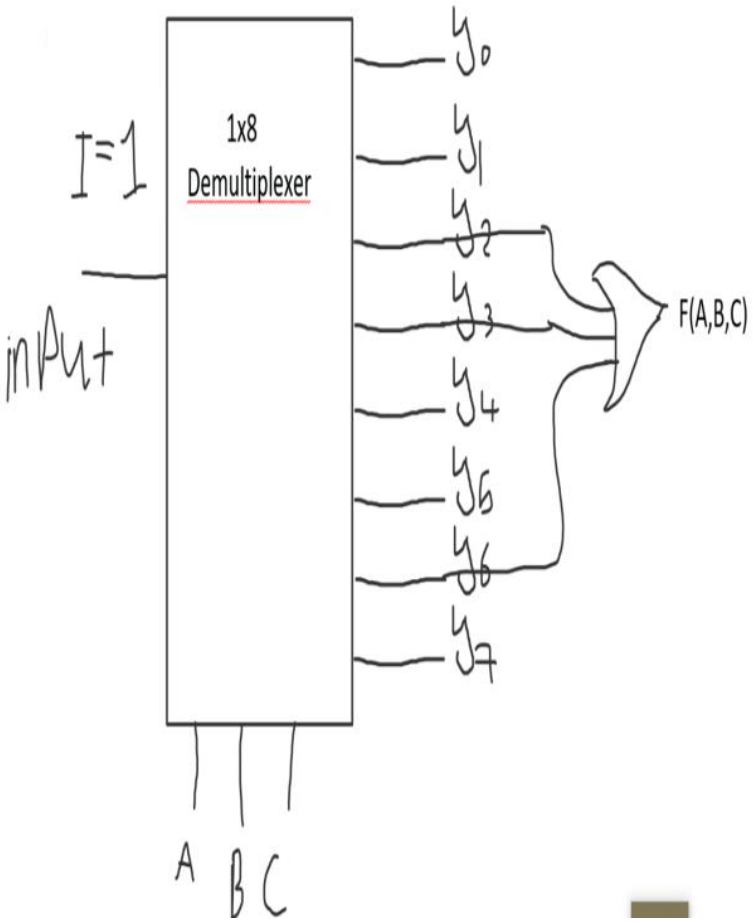
b) Sketch on Figure 3.1 the input connections necessary to implement the function in part (a). Observe that the inputs are connected to 0 or 1 depending on the value of the function for that min term.



Q2) Design a circuit which uses an SN74138 Demultiplexer to implement a sum- of-products expression, as follows: a) Convert the following expression into summation (Sum of Products –SOP-) form (i.e. F(A,B,C)=∑ (…)): $Y = f(A,B, C) = A'BC + BC'$

F=A'BC+ABC'+A'BC'

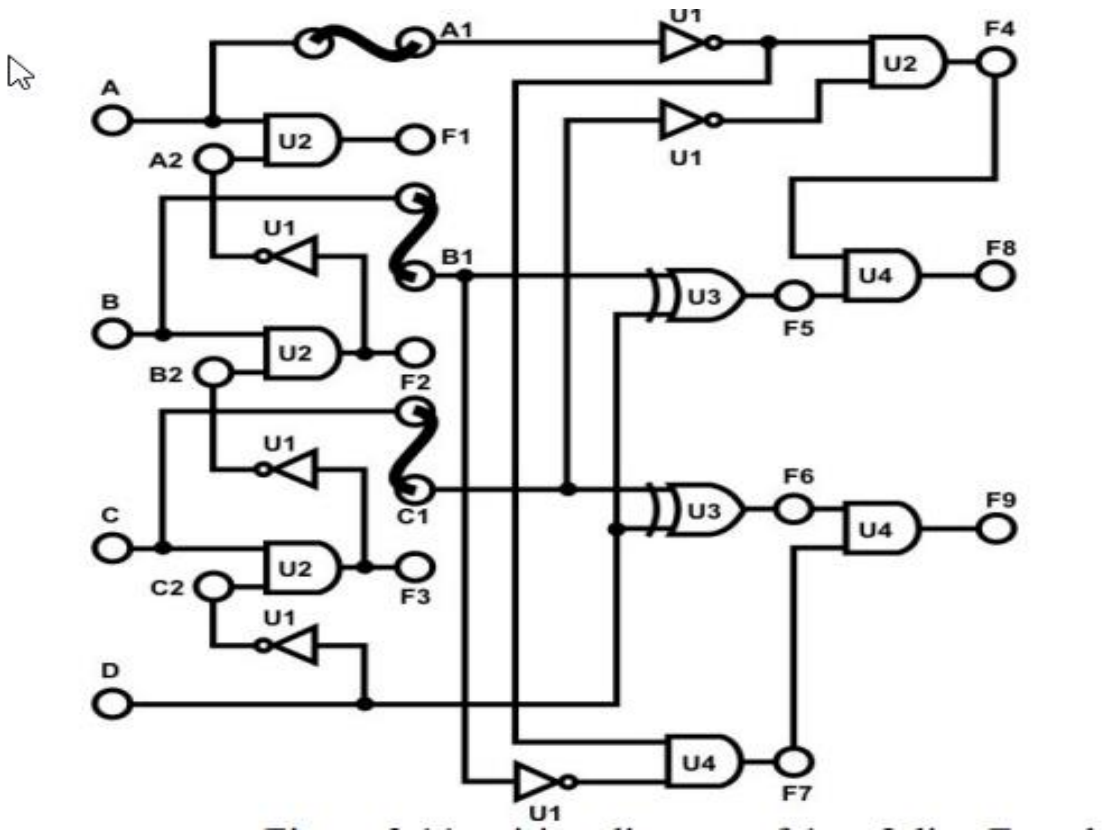
F= ∑(2,3,6)


1x8
Demultiplexer
I=1
input
y_0
y_1
y_2
y_3
y_4
y_5
y_6
y_7
F(A,B,C)
A
B
C

*Procedure, Data and results:

3.5.1 Constructing a 4-to-2 Encoder with Basic Gates:

At the beginning of every section, we inserted the +5 V and the ground



We inserted wires according to the figure above. Then we connected inputs and outputs. The results we received were:

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	1	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

According to the table above, it can be noticed that the encoder used was an active-high input-output encoder.

These connections and data collection took almost 15 minutes.

3.5.2 Constructing 9-to-4-Line Encoder with TTL IC:

we used The 74147 (U5) on block Encoder 2 of module IT-3004 in this experiment.

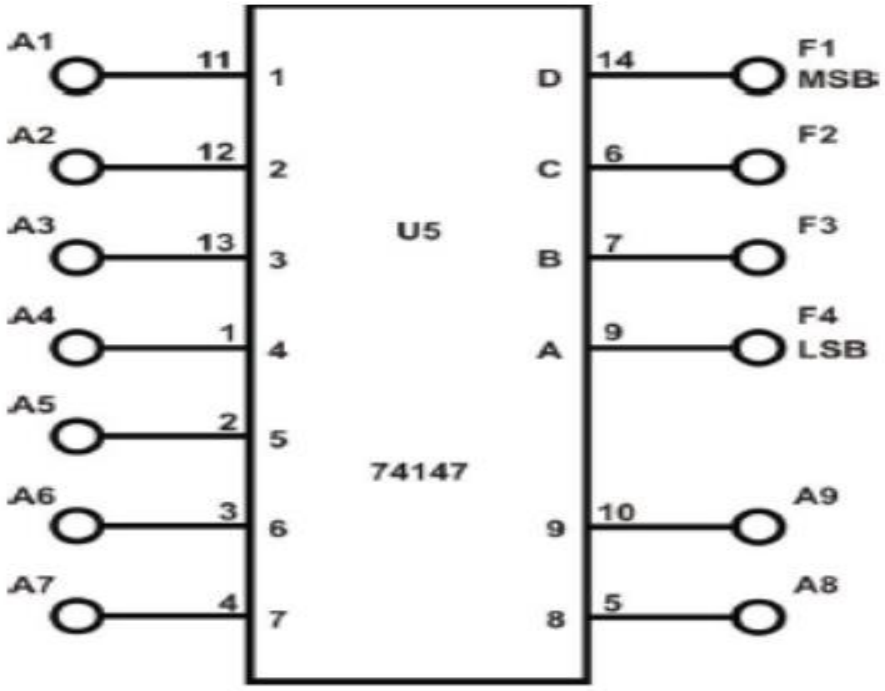


Figure 3.12: 74147 BCD Priority Encoder.

connect A1~A8 as inputs and s F1~F4 as outputs

Inputs									Outputs			
A9	A8	A7	A6	A5	A4	A3	A2	A1	F4(LSB)	F3	F2	F1(MSB)
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	1	1	0
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	1	0	0	1
1	0	0	0	0	1	1	1	1	0	1	1	0

these connections and data collection took almost 20 minutes.

3.5.3 Constructing 2-to-4 Line Decoder with Basic Gates:

we used Block Decoder 1 of module IT-3004 in this one.

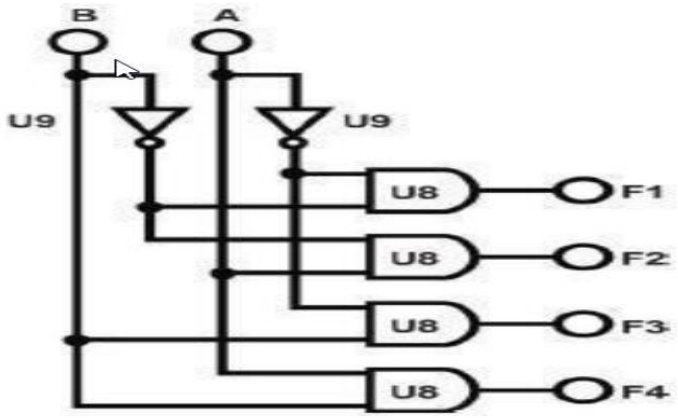


Figure 3.13: 2-to-4 Decoder.

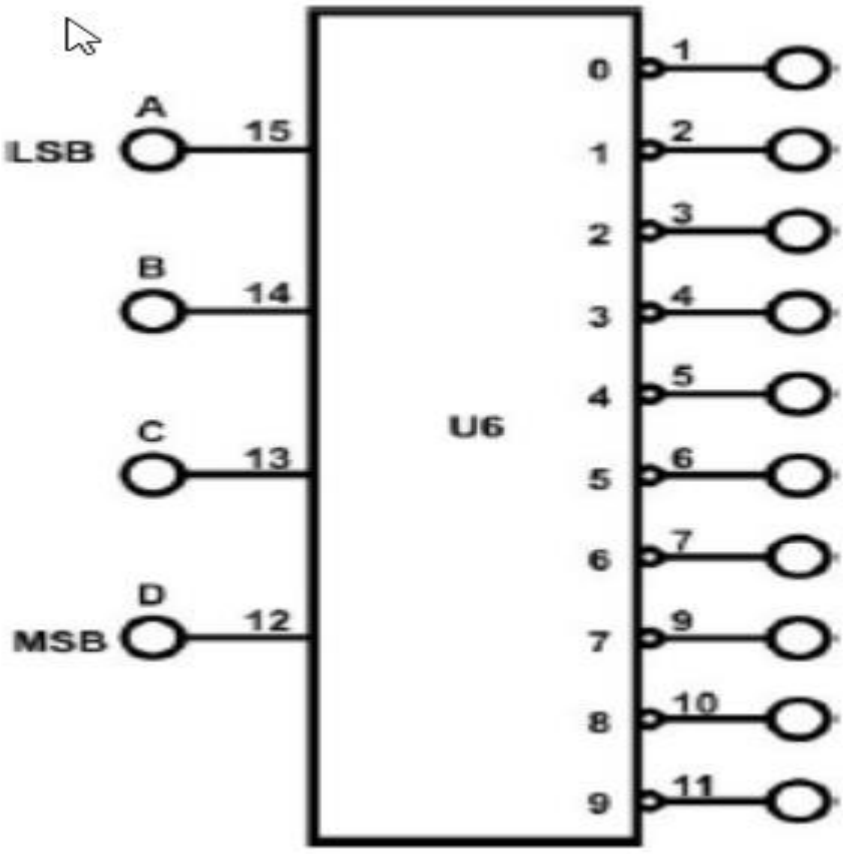
After Connecting inputs A, B to Data Switches SW0 and SW1. We Connected outputs F1~F4 to Logic Indicators L0~L3 respectively.

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

these connections and data collection took almost 10 minutes.

3.5.4 Constructing 4-to-10 Line Decoder with TTL IC:

In this section we used U6 (7442) on block Decoder 2 of module IT-3004. 7442 is a BCD-to-Decimal decoder IC. BCD: Binary Coded Decimal.



Connect inputs A-D to the Data Switches SW0-SW3, respectively. Then 10 outputs to corresponding Indicators L0~L9.

	Input				Output									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	○									
1	0	0	0	1		○								
2	0	0	1	0			○							
3	0	0	1	1				○						
4	0	1	0	0					○					
5	0	1	0	1						○				
6	0	1	1	0							○			
7	0	1	1	1								○		
8	1	0	0	0									○	
9	1	0	0	1										○

these connections and data collection took almost 25 minutes.

3.5.5 Constructing 2-to-1-Line Multiplexer with basic Gates:

Block Multiplexer 1 of module IT-3005 is used now as a 2-to-1 MUX.

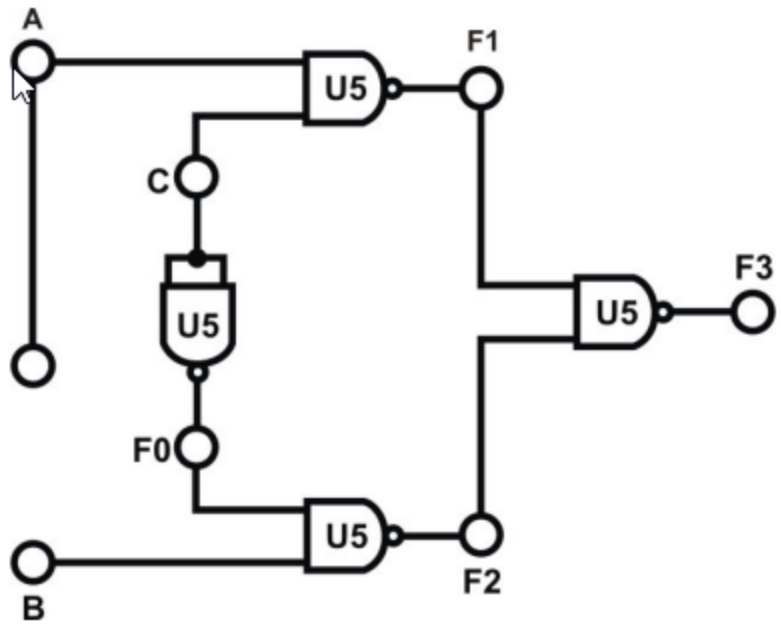


Figure 3.15: 2-to-1 Multiplexer.

inputs A, B to Data Switches SW0, SW1; Selector C to SW2. output F3 to Logic Indicator L0.

Inputs			Output
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

these connections and data collection took almost 10 minutes.

3.5.6 Constructing 8-to-1 Line Multiplexer with IC:

U3 (74LS151) on block Multiplexer 2 of module IT-3005 will be used in this section of the experiment

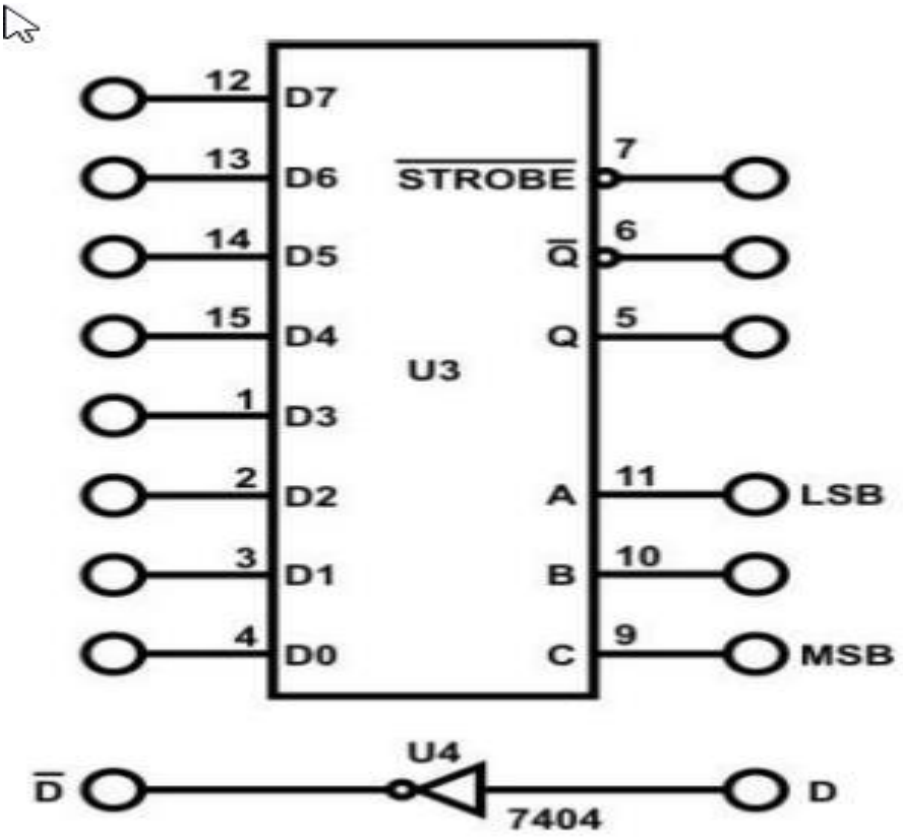


Figure (16): 8-to-1 MUX.

Connect inputs D0~D7 to DIP Switch 1.0~1.7; inputs C, B, A to Data Switches SW2, SW1, SW0.

Inputs			Output
C	A	B	Q
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

these connections and data collection took almost 20 minutes.

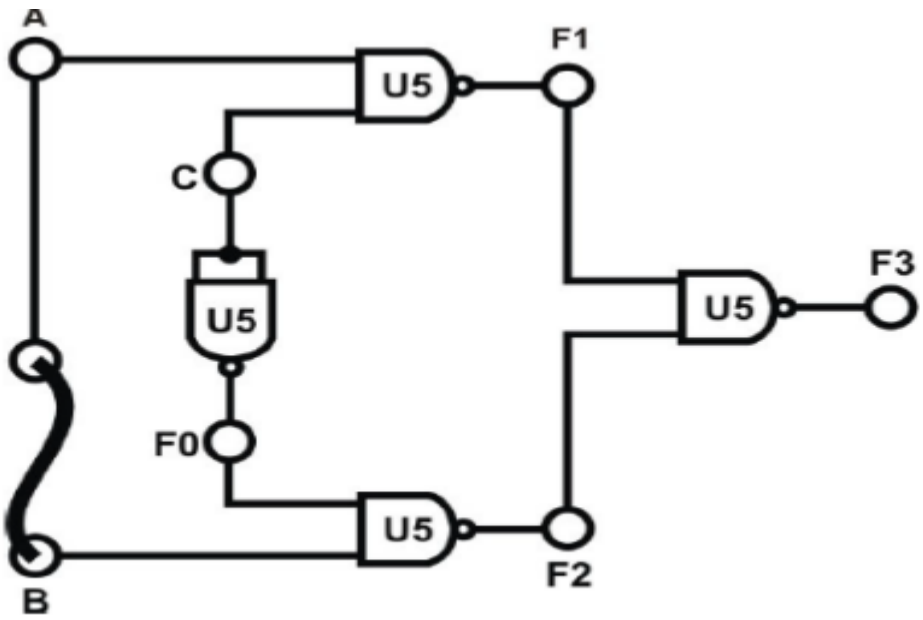
3.5.7 Using Multiplexer to implement a Logic Function Given the following function: F (A, B, C, D) =∑ (0,2,4,5,7,8,10,11,15).

Inputs				Output
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

these connections and data collection took almost 15 minutes.

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

Block Multiplexer 1 of module IT-3005 was the one to use. After following the connections in Figure 3.17. We connected A to Data Switch SW0; C to SW3; F1 and F2 to Logic Indicators L1 and L2 respectively.



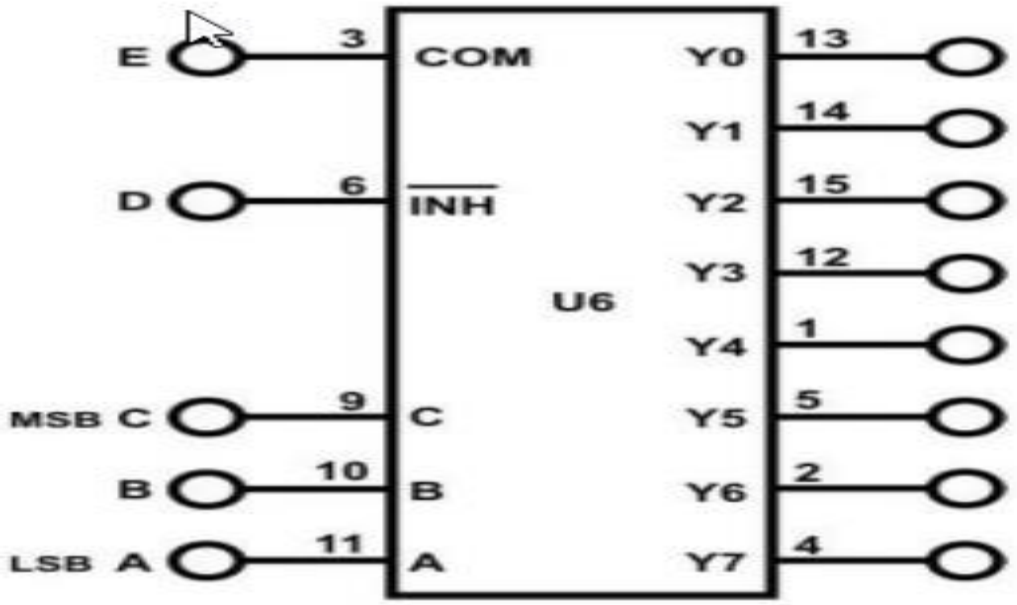
We Sat C to “0” and changed data at input A. then Sat C to “1”, change A and observe how F1 and F2 react to changes of A in the two cases of C.

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

these connections and data collection took almost 5 minutes

3.5.9 Constructing 1-to-8-Line Demultiplexer with CMOS IC:

In the last section we used U6 (4051) on block Demultiplexer of module IT-3005.



Connect E to DIP1.0; D to DIP1.1; A to SW0; B to SW1; C to SW2; outputs Y0~Y7 to Logic Indicators L0~L7 respectively

At D=0, apply the input sequence 1→0→1→0 to the common input E and observe outputs Y0~Y7. Did the outputs change as the input sequence is applied? Why?

Ans: yes, the sequence changes, since the D does the disable job and when its 0 the De_mux works regularly

and when its 1 the De-Mux will be disabled.

d) At D=1, apply the input sequence 1→0→1→0 to the common input E and observe outputs Y0~Y7. Did the outputs change as the input sequence is applied?

Ans: No, the sequence does not change, since the D does the disable job and when its 0 the De_mux works regularly and when its 1 the De-Mux will be disabled.

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

these connections and data collection took almost 25 minutes

* Discussion and Evaluation:

Through this experiment my mates and I discussed various stuff, one of them was about the Encoder that how could it hold more than one input, which is nonsense. Then we found out that the priority Encoder is the solution for this problem. We also could realize the relation between the output and the input at section 3.5.2 by finding that the digits were organized wrong.

And the outputs were low-active. In section 3.5.7 we could find the inputs for the multiplexer by finding some relations between inputs and outputs. In the last section we could had some help from the instructor to understand what does symbols D and E refers to and she helped us by telling us that D= disable and E=enable. And depending on the values of E and D the outputs will change or won't.

Thanks to her we could understand this section.

*Conclusions:

in this wonderful experiment I could deal with some pretty important ICs, which are multiplexers, Demultiplexers, Encoders, Decoders. All of the results we collected were similar to the theoretical part for all of the circuits we built. And due to the brilliant preparations my mates and I have made, we didn't face any connection errors.

*References:

https://www.allaboutcircuits.com/textbook/digital/chpt-9/decoder/

https://www.geeksforgeeks.org/encoder-in-digital-logic/

https://en.wikipedia.org/wiki/Multiplexer

We worked together and finished the experiment in almost 150 minutes.

Feedback:

this experiment is more interesting than the previous experiments. I think you must consider adding some sections that require connecting basic gates to build these ICs.