DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY

Dr. Anjad Badran

Experiment No. 3 - Encoders, Decoders, Multiplexers

Section 11

Islam Jihad Joma 1191375

T. Enas Jawabrah

14-4-2021

Abstract

Encoders, Decoders and Multiplexers are commonly used ICs in technology they provide the choice for the circuit to chose which function or circuit to use in every case, and the opposite for decoders and many other uses.

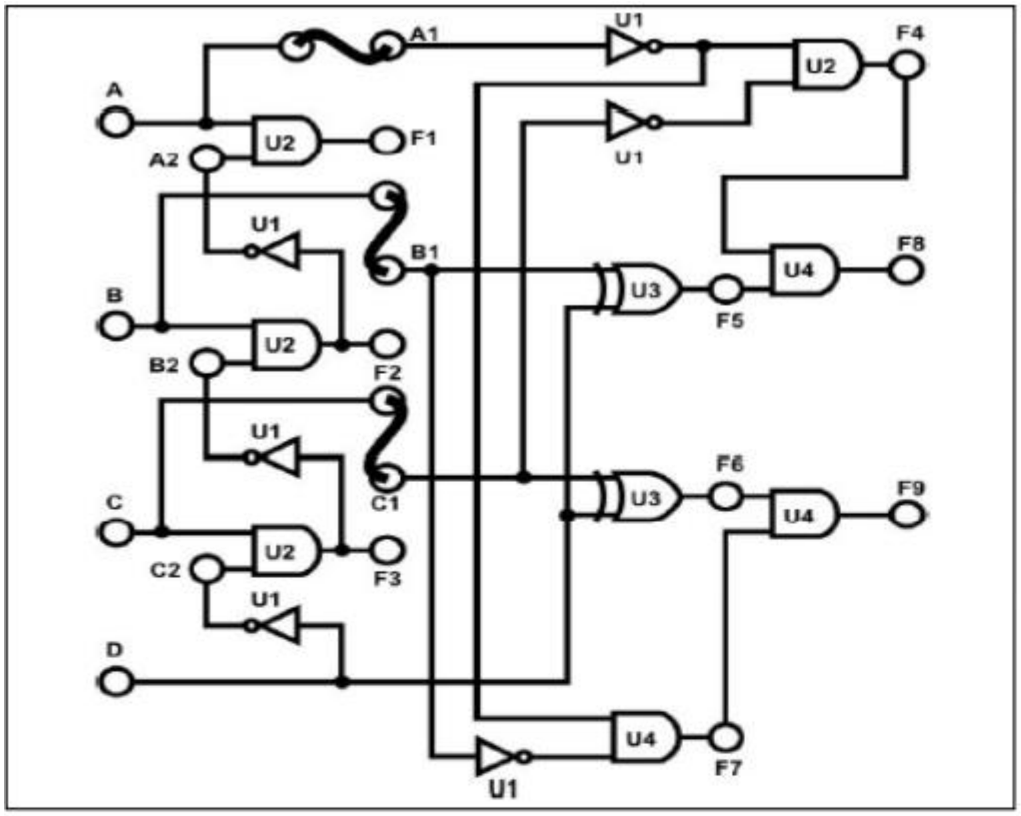
Table of content:

Introduction (Theory)

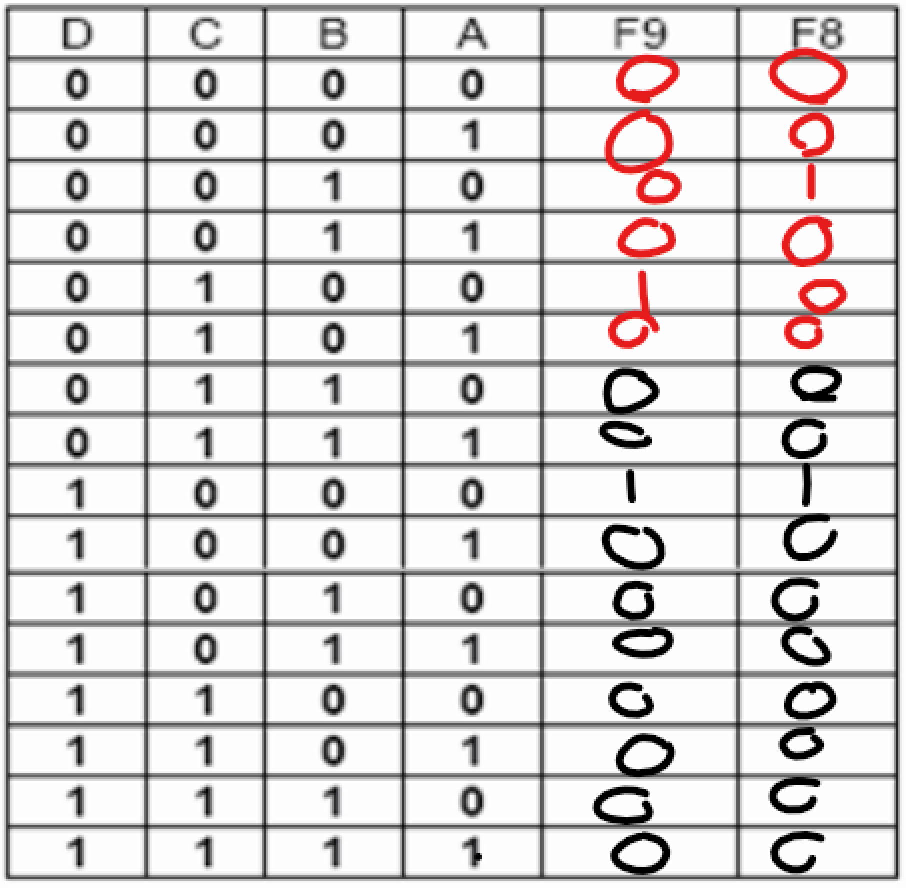
We have to create some circuits on the borders using the power supply on the lab to make Encoders, Decoders and Multiplexers, as the main point of the experiment. We saw how it work and how to implement it and what’s the output whenever are the inputs.

Procedure (Discussion & Results)

Constructing 4-to-2-Line Encoder with Basic Gates



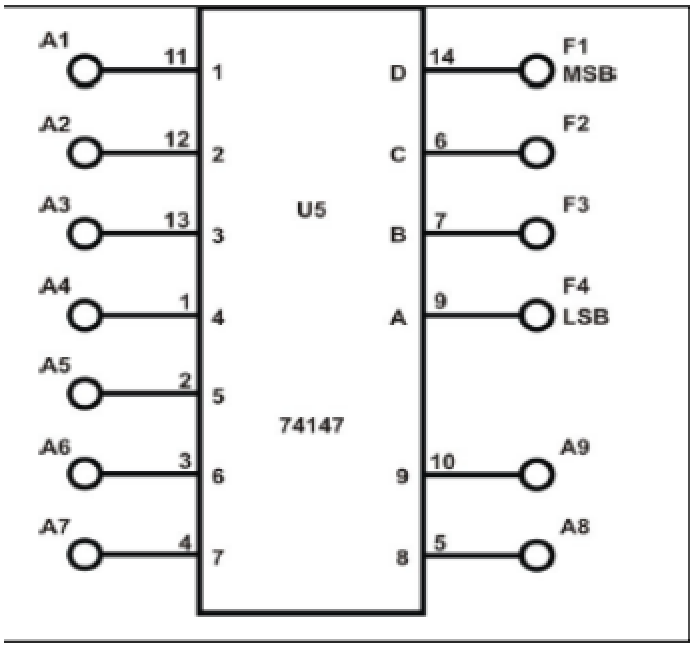
Here we constructed 4-2-line encoder by using basic gates (and, xor, not), it works as the following: where we find H in the inputs it turns into binary output for example if D was=1 only the output will be 11 but if there were more than 1 inputs carry H then the output will be 00, here is the truth table we created in the lab and you can see how the outputs looks whenever they are 00 or something normal:



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

Table 3.1

Constructing 9-to-4-Line Encoder with TTLIC



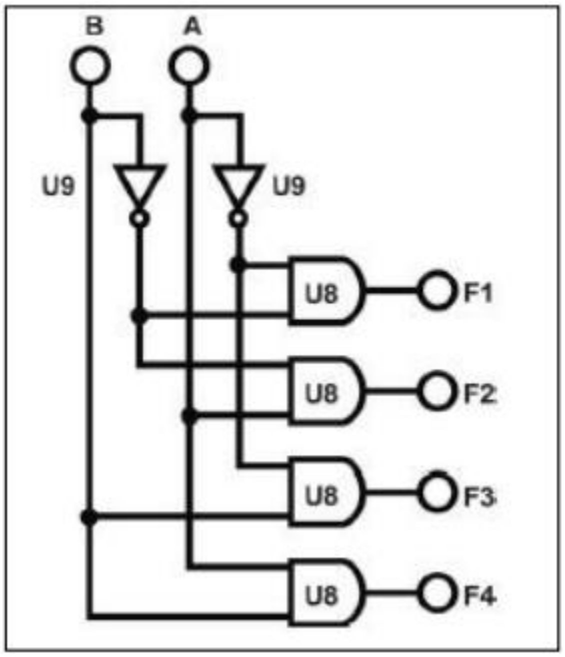
It’s the same as the previous circuit but with more inputs, the previous works for 4 inputs but this can contain 9 inputs and the output will be made of 4 bits

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

Table 3.2

As you can see the circuit work in active low method and give priority for the biggest input, for example the second row: the inputs are A9 A8 so the output is 0110 as it translates into 1001 because it’s act as active low, and it equal to 9 because the priority is for the biggest number and the 9th is bigger than 8th

Constructing 2-to-4 Line Decoder with Basic Gates

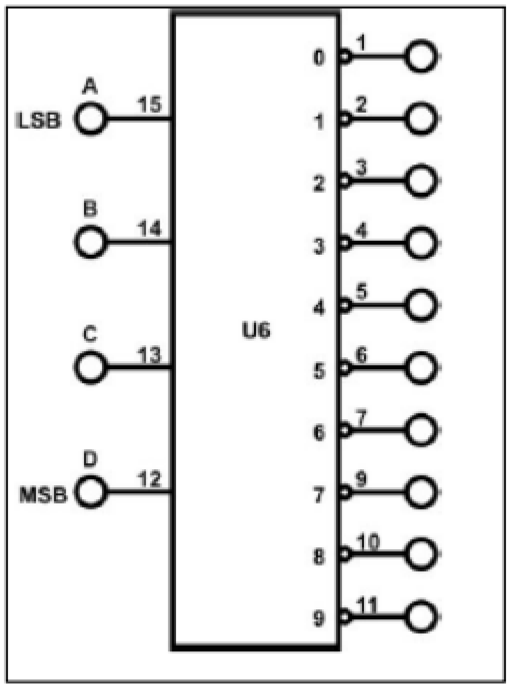


Here we constructed 2-4-line decoder by using basic gates (and, not), it works as the following: where we find H in the inputs it chose one output from F1-F4 output for example if A was=0 and B=0 only F1 will light. Here is the truth table we created in the lab and you can see how the outputs looks:

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 3.3

Constructing 4-to-10 Line Decoder with TTLIC



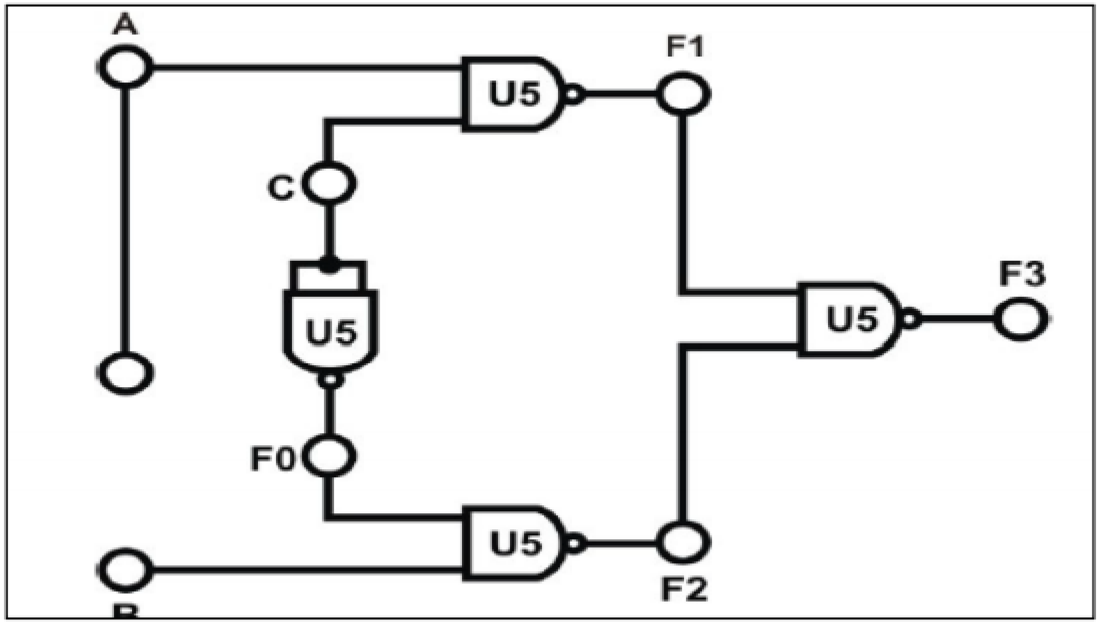
It's the same as the previous circuit but with more inputs, the previous works for 2 inputs (4 different choices) but this with 4 inputs (16 different choices but we have only 10 for this) and the output are one bit that lights or not. Here is the truth table:

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

Table 3.4

As you can see the circuit work in active High method, for example the second row: is 0001 which give the output for number 1 to light up

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

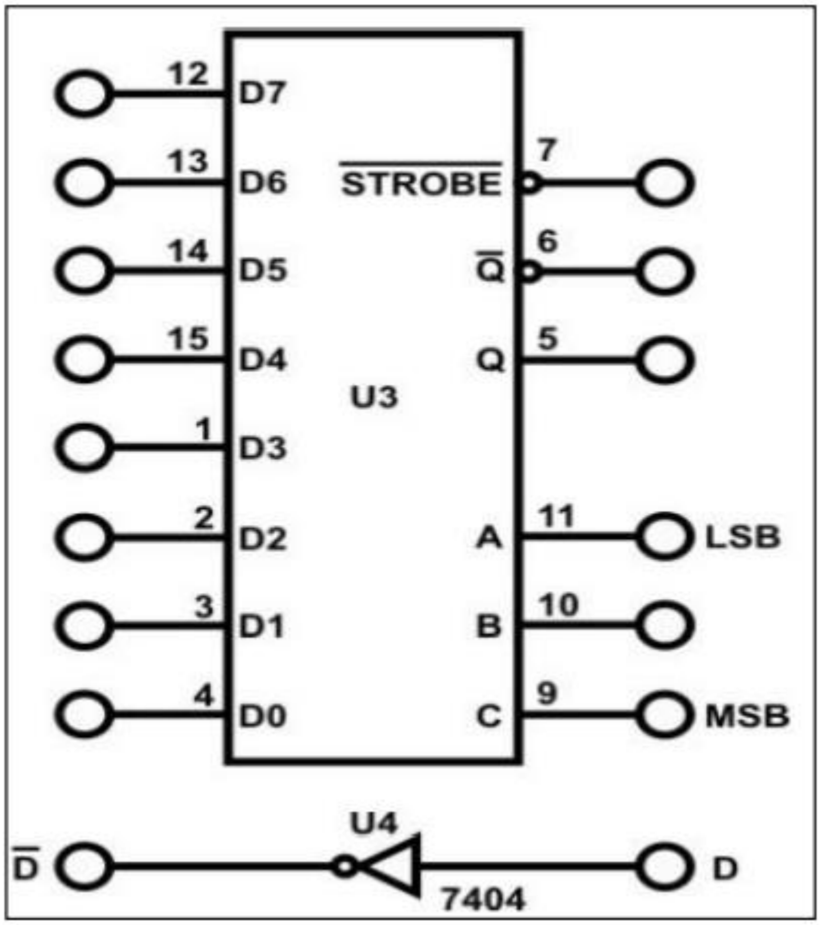


Here we have 3 inputs A, B and C the selector, this mux is made from basic gates and it follows this action, the selection decide what the output should be so if it was =0 then the output will be the same as the input B whatever A was and the opposite for C=1, here is the truth table that clarify it:

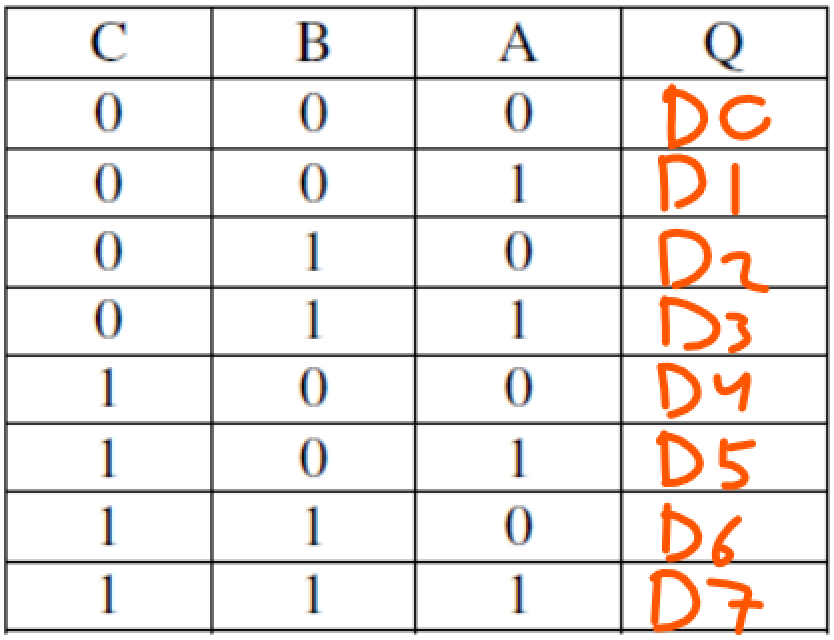
C	A	B	F3
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Table 3.5

Constructing 8-to-1 Line Multiplexer with IC



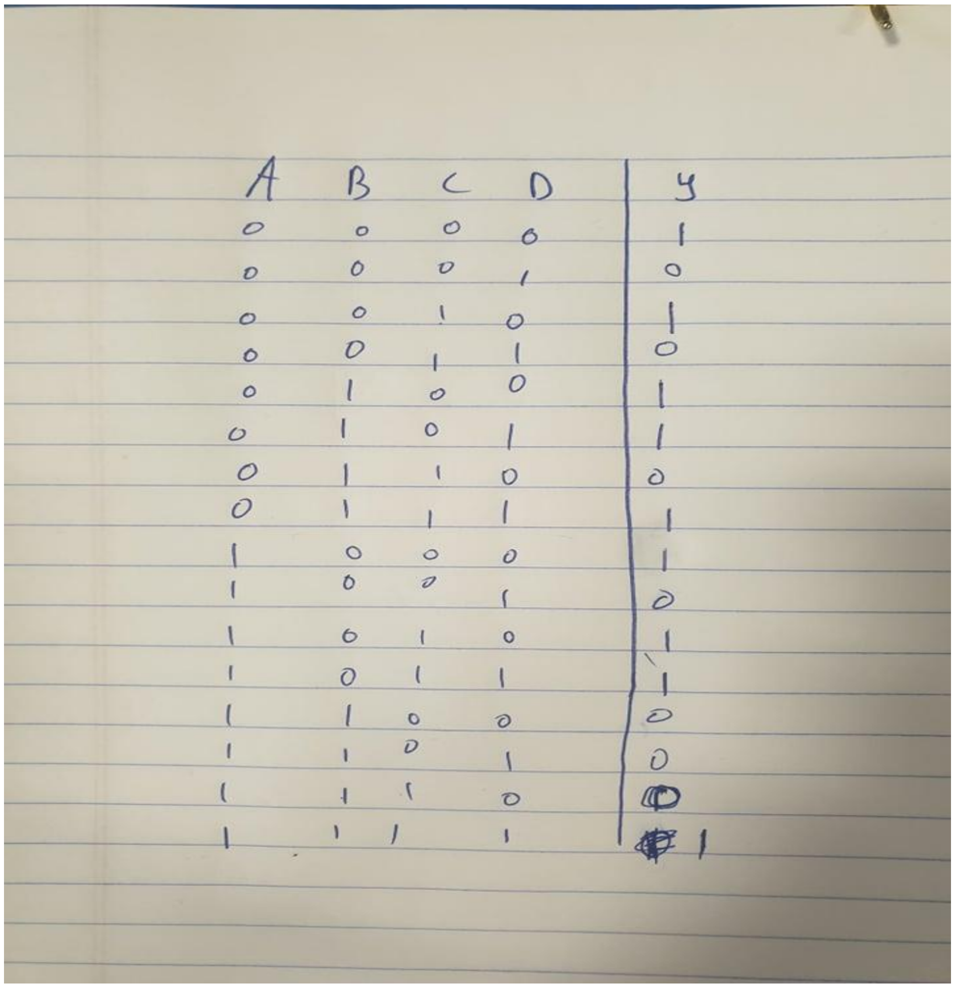
It’s IC mux that work for 8 different inputs and display the output in 3 binary bits, it works the same as the previous circuit but as it contain3 bits in the output we have to focus on the LSB & MSB to read the right outputs, here is the truth table:



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

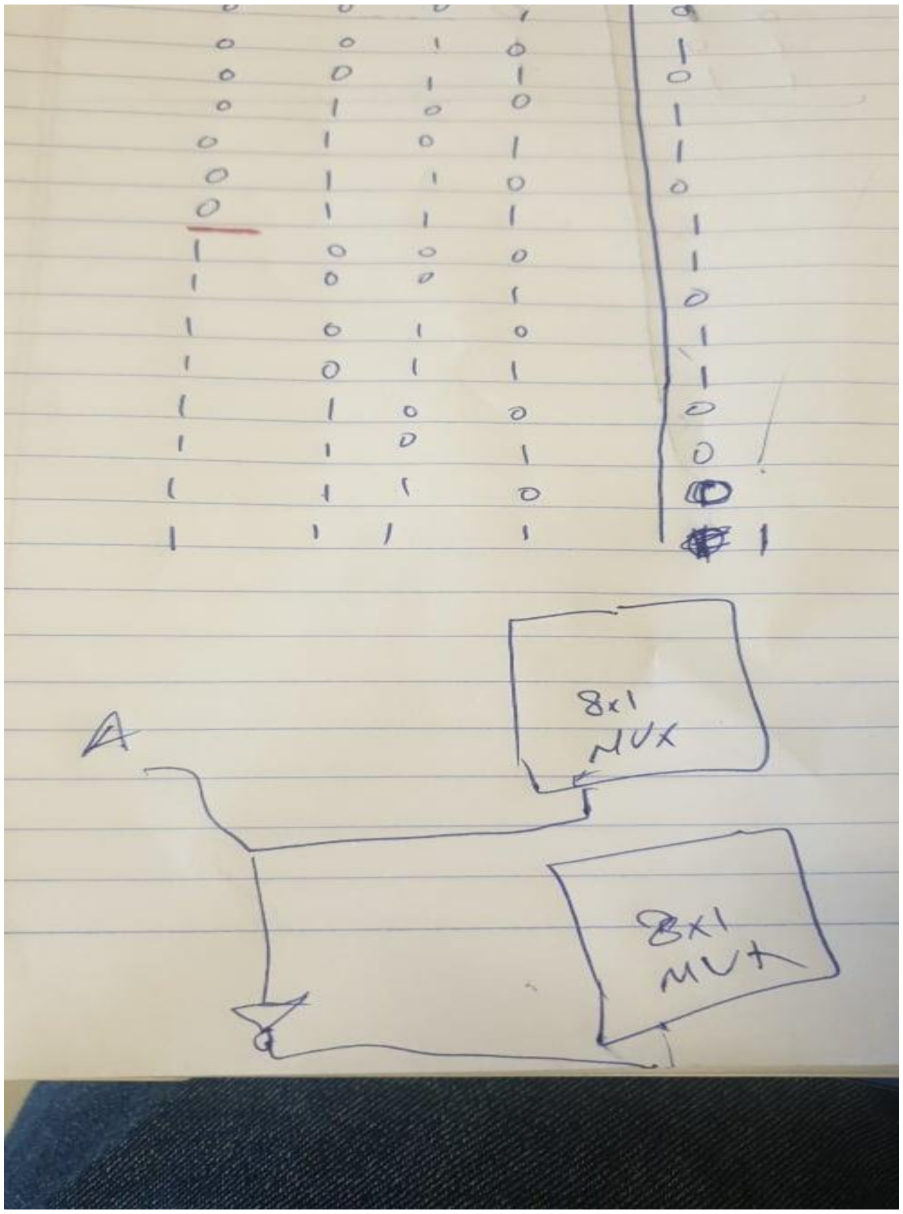
Table 3.6

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



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

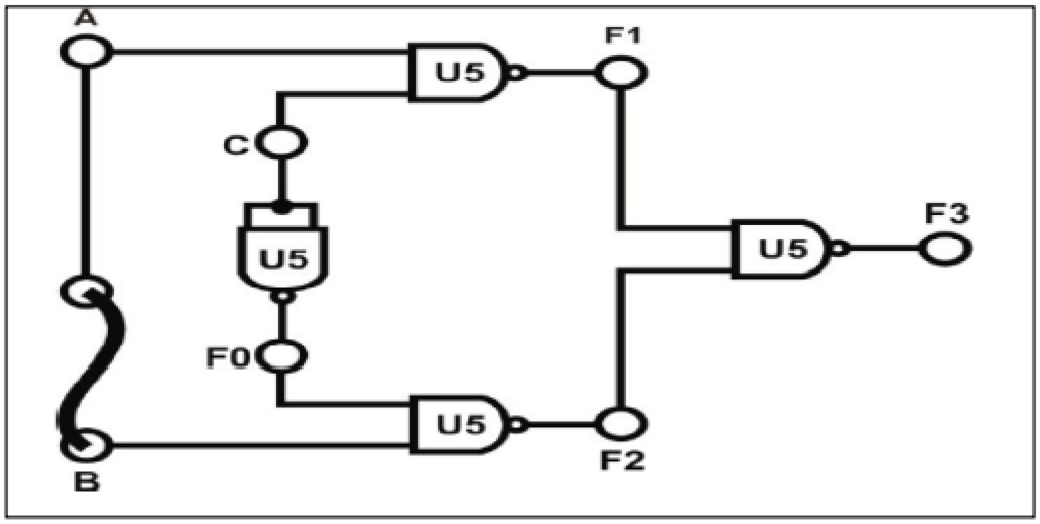
The work and connecting is the same as the previous circuit but here we have 4 inputs instead of 3, so the work is the same.

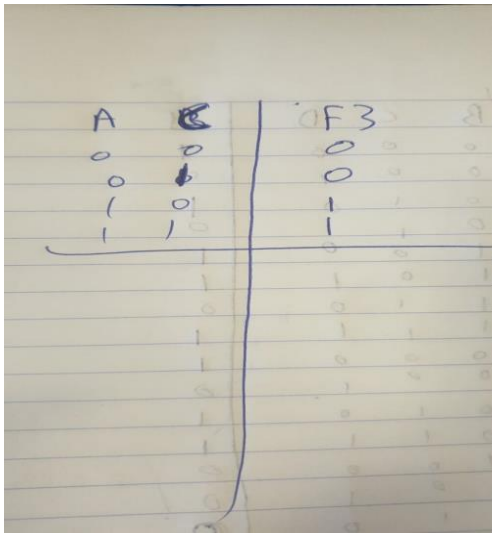


0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

We need a fifth bit with invertor to chose which mux to turn on and use.

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

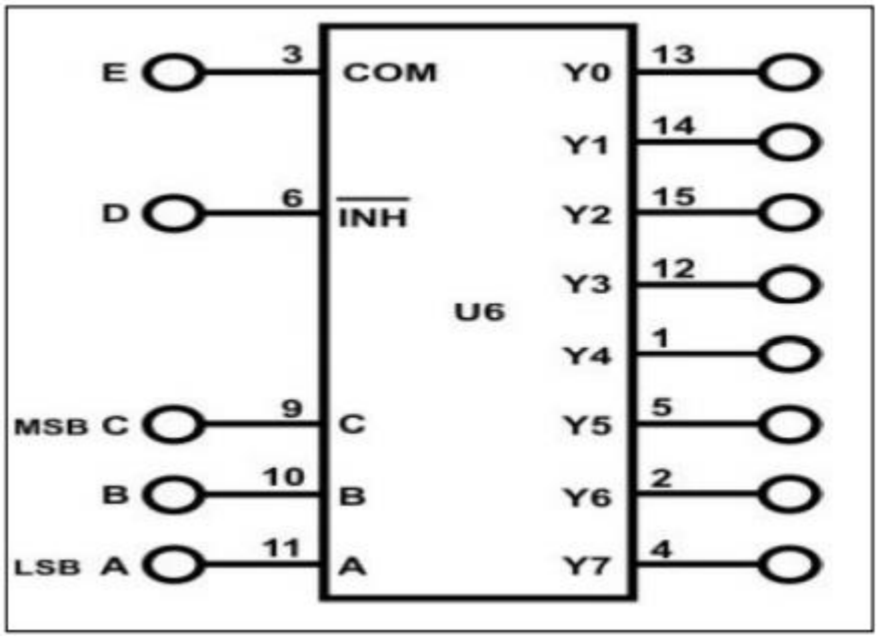




A	C	F3
0	0	0
0	1	0
1	0	1
1	1	1

As you see in the truth table the output is the same of the input A because the second input is connected with a wire with the first input ‘A’ so whatever the selection was it will lead to A

Constructing 1-to-8-Line Demultiplexer with CMOSIC



The D is for enable the circuit, we have 3 inputs and the output are 8 lines and to be carful of the LSB & MSB, the circuit is the same as the previous one and here is the truth table

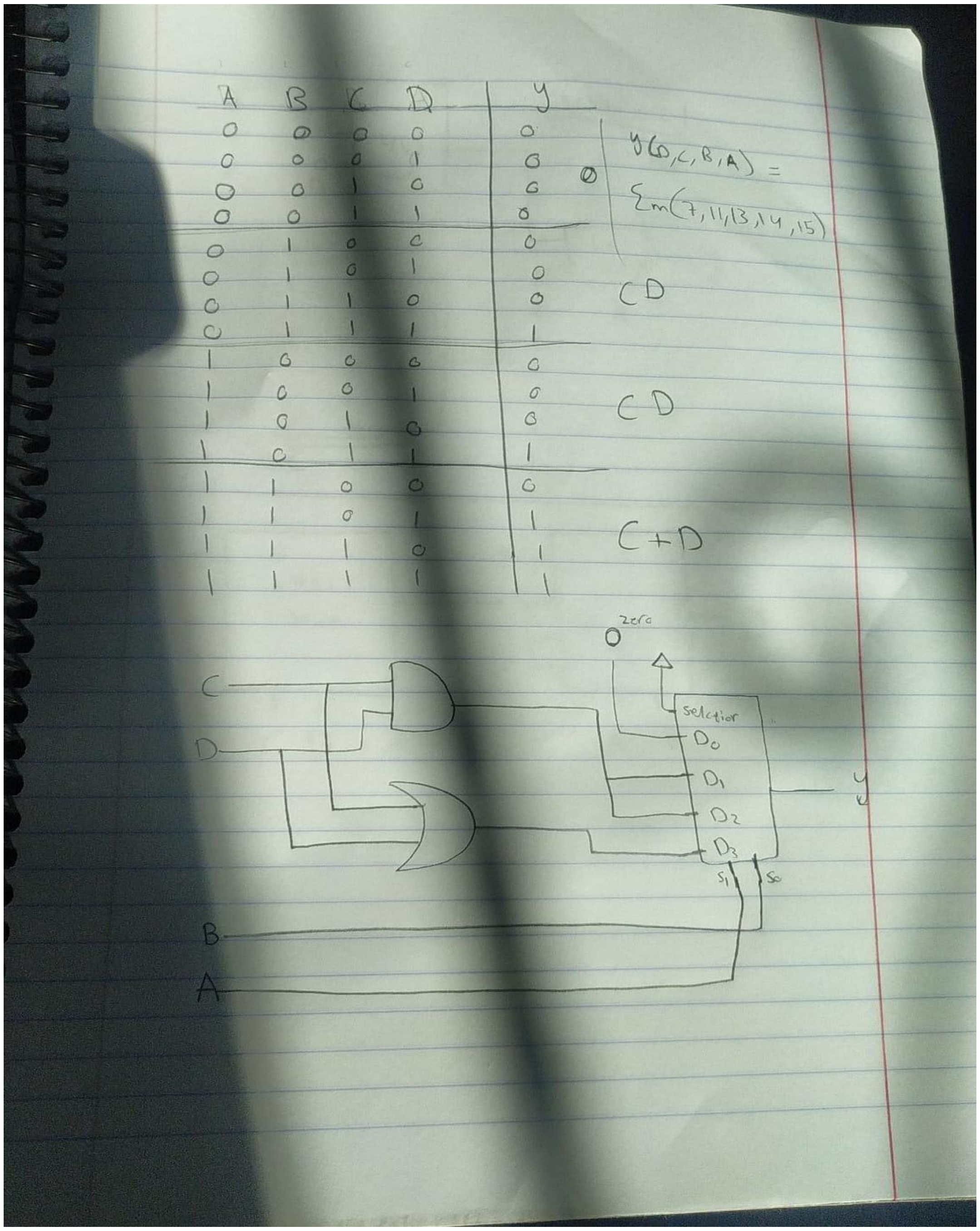
C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1							
0	0	1		1						
0	1	0			1					
0	1	1				1				
1	0	0					1			
1	0	1						1		
1	1	0							1	
1	1	1								1

Table 3.8

POST-LAB PROBLEM Solve the following problem in your report.

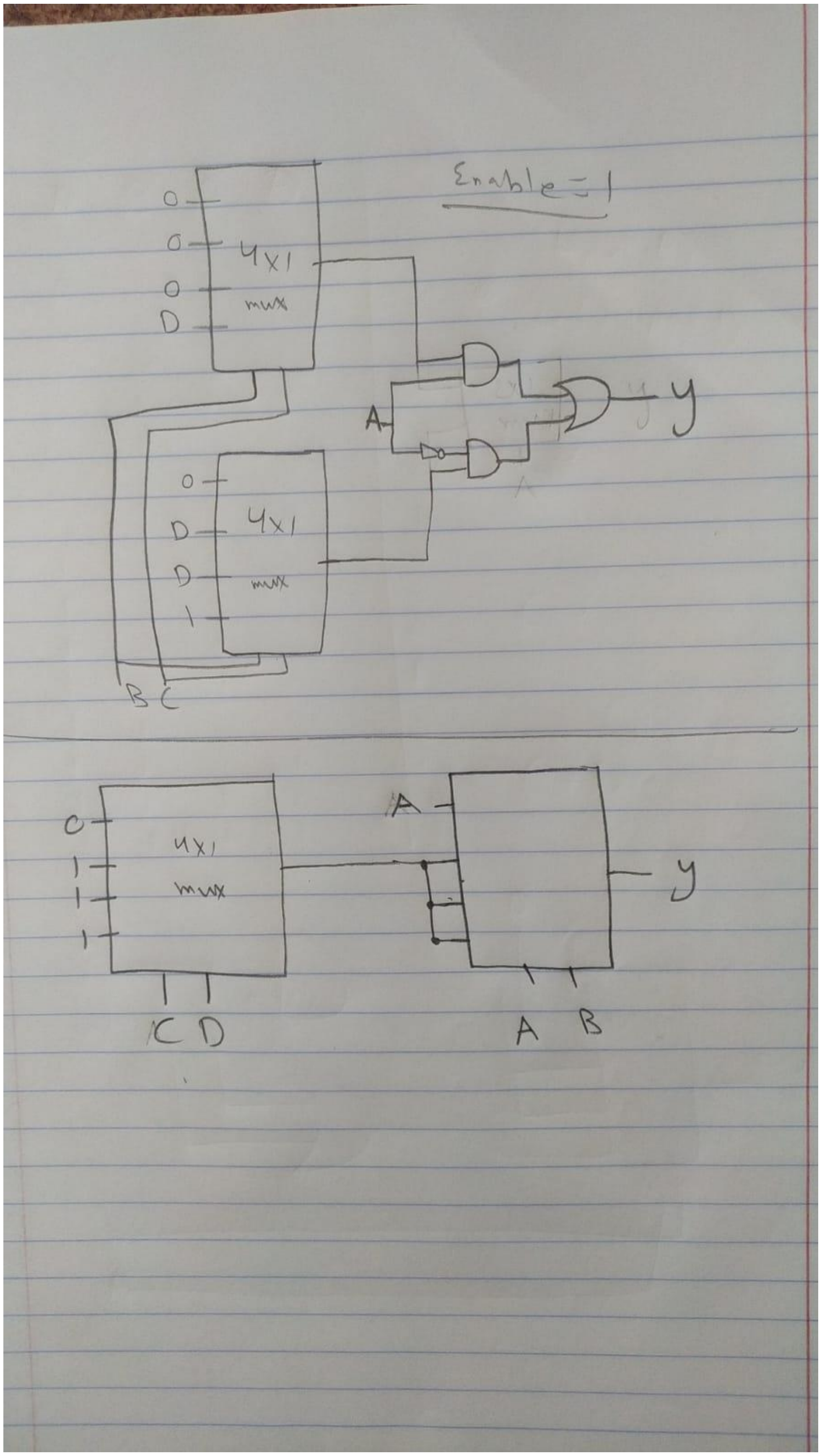
Design a Majority Circuit; a circuit that takes 4 inputs and 1 output, its output equals 1 when 3 or 4 of the inputs are 1. You can only use two 4×1 multiplexers.

I have 3 ways to solve the issuebut I'm not sure which one is the most correct



A	B	C	D	y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$y(D,C,B,A) = \sum m(7,11,13,14,15)$


Enable = 1
0
0
0
D
4x1
mux
A
y
0
D
D
1
4x1
mux
B C
C
1
1
1
4x1
mux
C D
A
y
A B

Conclusion

In this experiment I became familiar with Encoders, Decoders and Multiplexers and how to connect them together. And I have verified the correctness of my work by checking the circuit and it’s inputs & outputs more than once.

References

In this experiment I didn’t get help from outside the ALL Experiments PDF file, only my own work.

Appendix

I didn’t get help from outside the ALL Experiments PDF file, only my own work.