**Abstract**

This experiment aims to get us familiar with the understanding of different combinational circuits and how they work like encoders/decoders, multiplexers/demultiplexers by constructing their logic and monitoring output at different states, and also by using circuits like multiplexers to implement logical functions. The circuits shown are the ones implemented, in addition to the practical results of each one.

**Table of Contents**

1. Introduction ………………………………………………………………………………………………..........3
2. Encoders ……………………………………………………………………………………………………..........4
   1. A.Constructing4-to-2 Line Encoder with Basic Gates ……………………………………...4
3. Multiplexers ……………………………………………………………………………………………………….5
   1. Constructing 2-to-1 Line Multiplexer with basic Gates ……………………………………6
   2. Constructing 8-to-1 Line Multiplexer with IC …………………………………………………..7
   3. Using Multiplexer to Create a Logic Function ………………………………………………….8
4. Demultiplexers ……………………………………………………………………………………………………9
   1. Constructing 1-to-2 Line Demultiplexer with Basic Logic Gates ………………………9
   2. Constructing 1-to-8 Line Demultiplexer with CMOSIC …………………………………..10
5. Conclusion ………………………………………………………………………………………………………..11
6. References ………………………………………………………………………………………………………..13
7. Post-Lab problem ………………………………………………………………………………………….....14

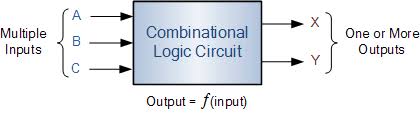
* **Note**: The circuits from pages 36-38 were not implemented due to a fault in the circuit board.

1. **Introduction**

Combinational Circuits (CC) are circuits made up of different types of logic gates. A logic gate is a basic building block of any electronic circuit. (1)

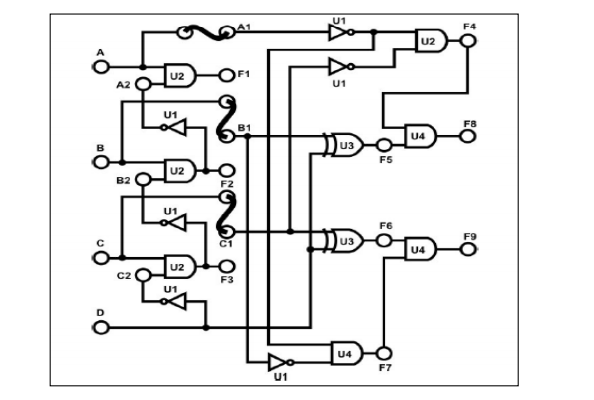
* The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
* The combinational circuit do not use any memory. The previous state of input does not have any effect on the present state of the circuit.
* A combinational circuit can have an n number of inputs and m number of outputs. (2)

Combinational circuits like Encoders, multiplexers and others will be discussed in detail further on.



1. **Encoders**

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, hence it encodes the information from 2^n inputs into an n-bit code. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2^n input lines with ‘n’ bits. (3)

* 1. **Constructing4-to-2 Line Encoder with Basic Gates**

**Figure (3.3)**

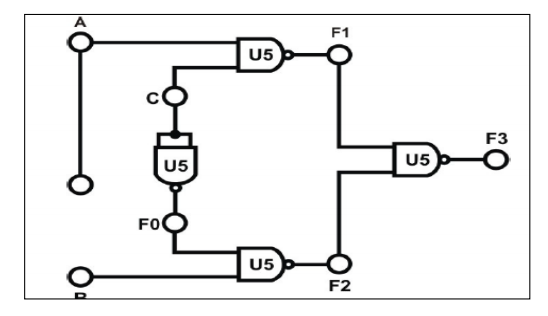
After constructing the circuit in fig 3.3 in the lab manual, the output (F9/F8) was as the following

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

1. **Multiplexers**

Multiplexers are combinational circuits designed to select one of multiple data inputs and produce a single output. They are commonly used in communication transmissions.

The input lines are selected depending on the selection inputs called control lines. The binary state of these inputs can either be low '0' or high '1'. Multiplexers have an even number of data input lines D as 2N with a corresponding number of control lines S.

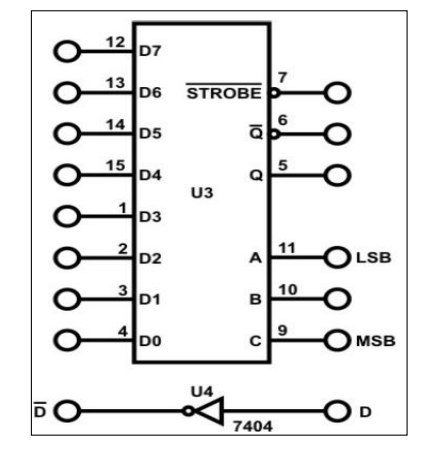
* 1. **Constructing 2-to-1 Line Multiplexer with basic Gates**

**Figure (3.7)**

The results of constructing the circuit in figure 3.7 were as the following

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

* 1. **Constructing 8-to-1 Line Multiplexer with IC**

****

**Figure (3.8)**

The output of this figure will be that on the input line selected, so for example, when the selection combination is “0,0” the output will be that on D0 whatever it is.

|  |  |  |  |
| --- | --- | --- | --- |
| C | B | A | Q (D=0, D=1) |
| 0 | 0 | 0 | 0 1 D0 |
| 0 | 0 | 1 | 0 1 D1 |
| 0 | 1 | 0 | 0 1 D2 |
| 0 | 1 | 1 | 0 1 D3 |
| 1 | 0 | 0 | 0 1 D4 |
| 1 | 0 | 1 | 0 1 D5 |
| 1 | 1 | 0 | 0 1 D6 |
| 1 | 1 | 1 | 0 1 D7 |

* 1. **Using Multiplexer to Create a Logic Function**

F(A, B, C, D)=∑(0,2,4,5,7,8,10,11,15)

D’

D’

1

D

D’

1

0

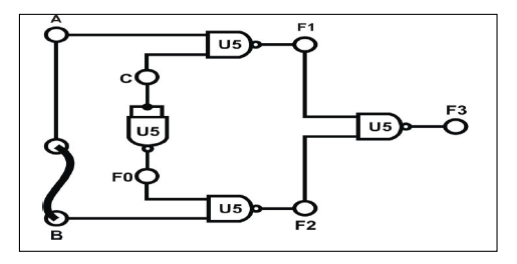
D

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | F |
| 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 |

1. **Demultiplexers**

The De-multiplexer does the reverse operation of the multiplexer. This means that it receives a single data input and depending on the selection of its control lines it produces multiple outputs. It is also referred to as a data distributor. It converts a single serial input into parallel data outputs on the output line. Figure 8 is a block diagram representation of the De-multiplexer.

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

****

**Figure (3.10)**

Implementing the circuit shown in figure 3.10 in the manual resulted in the following

When C =”0”

A = 0 L1, L0 = 1

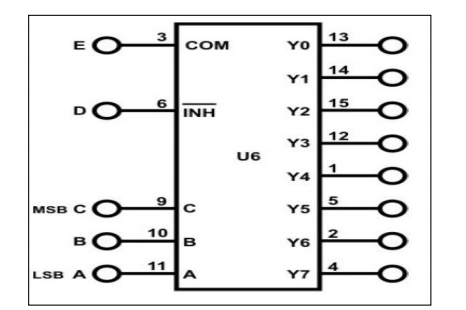
A = 1 L0 = 1

When C = “1”

A = 1 L1 = 1

A = 1 L1, L0 = 1

* 1. **Constructing 1-to-8 Line Demultiplexer with CMOSIC**

****

**Figure (3.11)**

The results of implementing figure 3.11 are shown in the table below

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

1. **Conclusion**

The outcome of this experiment was getting to know in detail what combinational circuits are and how each component works, and also that they can be implemented in different ways using various logic gates like NAND and NOR etc.

These circuits and circuit components are the basis of mainly all digital design, which shows the importance of understanding what each one does in order to be able to use them in future projects and make the most out of their specifications and functionalities.

* Encoders

An encoder is a combinational circuit that converts binary information in the form of a 2N input lines into N output lines, which represent N bit code for the input. For simple encoders, it is assumed that only one input line is active at a time. (4)

* Decoders

A decoder does the opposite job of an encoder. It is a combinational circuit that converts n lines of input into 2n lines of output.

* Multiplexers

Combinational circuits which have many data inputs and single output depending on control or select inputs.​ For N input lines, log n (base2) selection lines, or we can say that for 2n input lines, n selection lines are required. Multiplexers are also known as “Data n selector, parallel to serial convertor, many to one circuit, universal logic circuit​”. Multiplexers are mainly used to increase amount of the data that can be sent over the network within certain amount of time and bandwidth. (5)

* Demultiplexers

A demultiplexer (or demux) is a device that takes a single input line and routes it to one of several digital output lines. A demultiplexer of 2n outputs has n select lines, which are used to select which output line to send the input. (6)

Basically meaning it does the opposite of multiplexers.

**6. references**

(1) <https://study.com/academy/lesson/basic-combinational-circuits-types-examples.html>

(2) <https://www.tutorialspoint.com/computer_logical_organization/combinational_circuits.htm>

(3) <https://www.geeksforgeeks.org/digital-logic-encoder/>

(4) <https://www.geeksforgeeks.org/digital-logic-encoders-decoders/>

(5) <https://www.geeksforgeeks.org/multiplexers-digital-electronics/>

(6) <http://electronics-course.com/demux>

**7. Post Lab Problem**

Design a majority circuit that takes 4 input and 1 output. Its output equals 1 when 3 or 4 of the inputs are 1. Only use two 4X1 multiplexers.

Truth Table

0

0

0

D

0

D

D

1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | F |
| 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 |

Design