

**AMERICAN INTERNATIONAL**

**UNIVERSITY BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY DEPARTMENT OF**

**ELECTRICAL AND ELECTRONICS ENGINEERING**

**Spring 2024-2025**

**DIGITAL LOGIC AND CIRCUITS LABORATORY**

**Section: D Group: 03**

**LAB REPORT NAME:**

# Designing Multiplexer (MUX) and De-multiplexer (DEMUX), Priority Encoder and Decoder Circuits

**Supervised By:**

**S M TANVIR HASSAN SHOVON**

**Submitted By:**

|  |  |  |
| --- | --- | --- |
| **SL** | **Student ID** | **Student Name** |
| **1.** | 23-50856-1 | Basudeb Kundu |

**Submition Date:18.04.2025**

# Title:

# Designing Multiplexer (MUX) and De-multiplexer (DEMUX), Priority Encoder and Decoder Circuits.

**Theory and Methodology:**

**Part I: Multiplexer and De-multiplexer**

A multiplexer (or mux) is a device that selects one of several inputs and forwards the selected input into a single line. A multiplexer of 2*n* inputs has *n* selection lines, which are used to select which input has to be sent to the output. A multiplexer is also called a data selector.

A de-multiplexer (or de-mux) is a device taking a single input and selecting one of many data-output- lines, which is connected to the single input.

**Multiplexer:**

n a computer system, selecting data from one of multiple sources is often necessary. Suppose we have two data sources, provided as input signals **I0** and **I1**. These signals can change over time, possibly at regular intervals. We aim to design a circuit that outputs the same value as either **I0** or **I1**, depending on the value of a single selection pin **S**.

Since we have **one selection pin**, there are **two possible combinations**:

* If **S = 0**, the output **Y = I0**
* If **S = 1**, the output **Y = I1**

The relationship between the number of inputs and selection pins follows a general rule:  
If a multiplexer has **n selection pins**, it can handle **2ⁿ inputs**. A **2-to-1 MUX** has **one selection pin** and **two inputs**. Similarly, other multiplexers include **4-to-1**, **8-to-1**, and so on.

# Table: 1

|  |  |
| --- | --- |
| **S** | **Y** |
| 0 | I0 |
| 1 | I1 |

From the truth table, we derive the Boolean expression for the output:

Y = S̅ I0 + S I1

This equation represents the logic function of a **2-to-1 multiplexer**, where the selection pin determines which input is passed to the output.

The logic circuit of the equation (1) is given in figure 1.

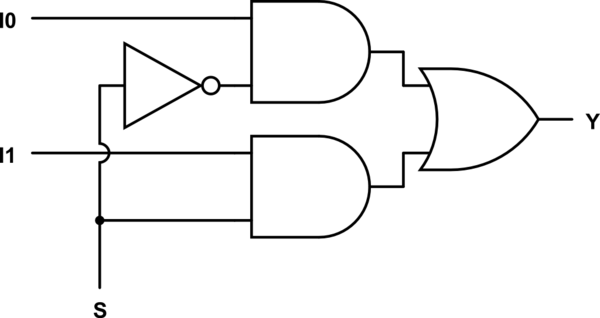


Fig. 1: 2 to 1 Multiplexer

# De-multiplexer:

A **demultiplexer (DEMUX)** is the opposite of a multiplexer. It takes a **single input** and routes it to one of **multiple outputs** based on the value of selection pins. The remaining outputs remain **low** (inactive). Depending on the number of outputs, demultiplexers are categorized as **1-to-2**, **1-to-4**, **1-to-8**, etc. If a demultiplexer has **n selection pins**, it can have a maximum of **2ⁿ outputs**.

For a **1-to-2 DEMUX**, we have:

* **One data input (Din)**
* **One selection pin (S)**
* **Two outputs (I0 and I1)**

The selection pin determines which output receives the input signal **Din**, while the other remains **low**.

The truth table for a **1-to-2 DEMUX** is as follows:

|  |  |  |
| --- | --- | --- |
| **S** | **Y0** | **Y1** |
| 0 | Din | 0 |
| 1 | 0 | Din |

From this truth table, the Boolean expressions for the outputs are:

Y0 = S̅ .Din

Y1=S⋅Din

This represents the logic function of a **1-to-2 demultiplexer**, where the selection pin determines which output receives the input signal.

A diagram of a circuit

AI-generated content may be incorrect.

**Fig. 2:** 1-to-2 line De-multiplexer

# Part II: Encoder and Decoder:

An encoder is a device or a circuit that converts information from one format or code to another. A decoder does the reverse operation of the encoder. It undoes the encoding so that the original information can be retrieved. Both the encoder and decoder are combinational circuits.

Encoding and decoding are very widely used ideas. They have applications in electronic circuits, software programs, medical devices, telecommunication and many others. In this experiment, a very basic 2-to-4 line decoder and a decimal to BCD encoder will be constructed.

A decoder can convert binary information from n input lines to a maximum of 2*n* unique output lines. The 2-to-4 line decoder will take inputs from two lines and convert them to 4 lines.

A B

D0

D1

D2

D3

**Fig. 3:** 2-to-4 line decoder

The expressions for implementing 2-to-4 line decoder are

|  |  |
| --- | --- |
| 𝐷0 = 𝐴̅𝐵̅ | (6) |
| 𝐷1 = 𝐴̅𝐵 | (7) |
| 𝐷2 = 𝐴𝐵̅ | (8) |
| 𝐷3 = 𝐴𝐵 | (9) |

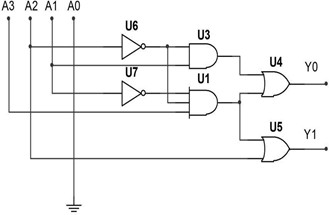
Truth table for 2-to-4 line decoder is given below –

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | D0 | D1 | D2 | D3 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

# Priority encoder:

A priority encoder is a [circuit](http://en.wikipedia.org/wiki/Electronic_circuit) or [algorithm](http://en.wikipedia.org/wiki/Algorithm) that compresses multiple [binary](http://en.wikipedia.org/wiki/Binary_code) inputs into a smaller number of outputs. The output of a priority encoder is the binary representation of the original number starting from zero of the most significant input bit. They are often used to control [interrupt requests](http://en.wikipedia.org/wiki/Interrupt_request) by acting on the highest priority request. If two or more inputs are given at the same time, the input having the highest priority will take [precedence.](http://en.wiktionary.org/wiki/precedence)

In this experiment a 4-to 2 priority encoder with a priority sequence of 2,1,3,0 has been shown. It means, in this priority encoder 2 has the highest priority and 0 has the lowest. If 2 is high then other numbers are ignored (even if any of them are high at the same time) and output would be binary representation of 2, i.e., Y1Y0=10. If 2 is found to be low, then next priority is given to 1. So, in this case if 1 is high, then 3 and 0 are ignored and output will be binary representation of 1, i.e., Y1Y0=01 and so on.



**Fig. 4:** 4-to 2 priority encoders with a priority sequence of 2,1,3,0

The expressions for implementing the above priority encoder–

Y0 = A2’.A1 + A3.A2’.A1’ Y1 = A2 + A3.A2’.A1’

Truth table for this priority encoder is given below –

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A3 | A2 | A1 | A0 | Y1 | Y0 |
| x | 1 | x | x | 1 | 0 |
| x | 0 | 1 | x | 0 | 1 |
| 1 | 0 | 0 | x | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 |

# Introduction:

In this experiment, students will learn how to design and implement multiplexers (MUX) and demultiplexers (De-MUX) of different sizes using basic logic gates. They will also learn how to construct bigger multiplexer using smaller multiplexers. Students will also construct encoder and decoder circuits. Encoder and decoder circuits are very useful in information transmission, conversion, compression and maintaining the secrecy of any information.

## **Apparatus:**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | NOT Gate - | IC 7404 | 1 [pc] |
| 2. | AND Gate - | IC 7408 | 1 [pc] |
| 3. | OR Gate - | 5 input OR | 1 [pc] |
|  |  | 4 input OR | 2 [pcs] |
|  |  | 2 input OR | 1 [pc] |
|  |  |  |  |

# Precautions:

1. It should be ensured that all the LEDs and toggle switches on the trainer board are functioning properly.

2. Short circuits should be avoided, as heat may be generated due to high current flow, which can be harmful to the components.

# Experimental Procedure:

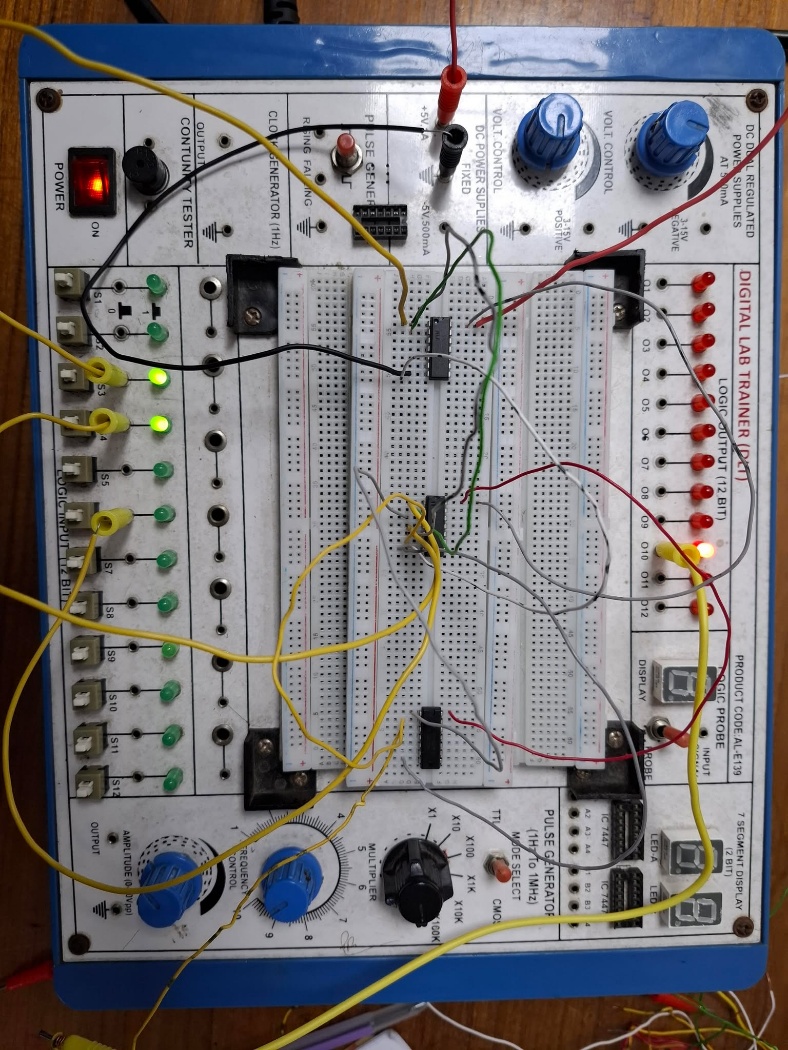
1. The circuit should be connected according to the provided figures.

2. The toggle switches on the trainer board should be used to provide input signals to the circuits, and the outputs should be connected to the LEDs on the trainer board.

3. The input signals should be applied, and the corresponding output signals should be observed and recorded.

# Measurement:

# Fig. 1: 2 to 1 Multiplexer :-

****

**A close-up of a circuit board

Description automatically generated**

**Fig. 2:** 1-to-2 line De-multiplexer

A circuit board with wires and switches

Description automatically generated

**Fig. 3:** 2-to-4 line decoder

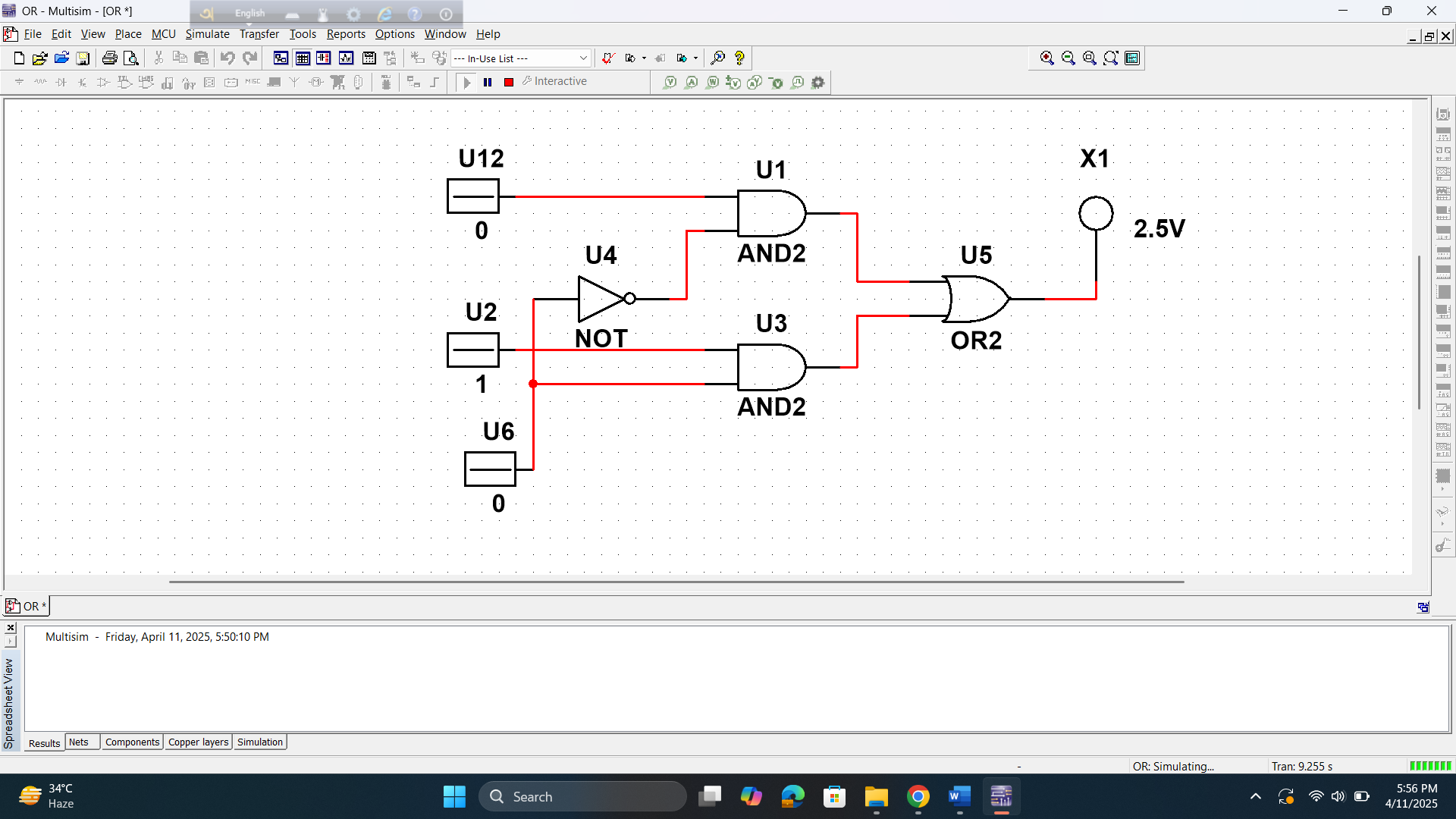
**A close-up of a circuit board

Description automatically generated**

**A close-up of a circuit board

Description automatically generated**

# Simulation and Measurement:



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1-to-2 line De-multiplexer

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

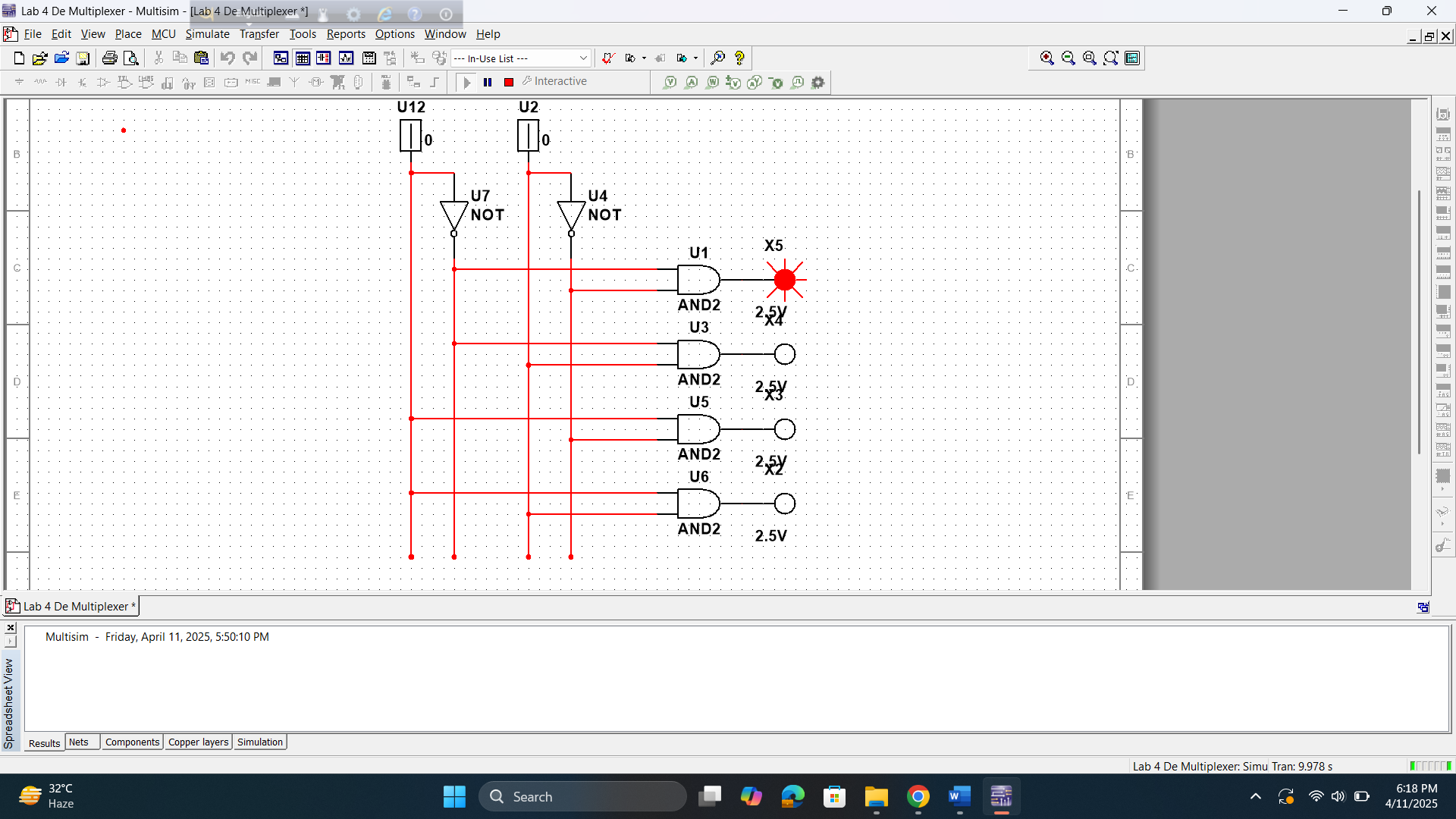
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

2-to-4 line decoder



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Discussion and Conclusion:**

The MUX, DEMUX, Priority Encoder, and Decoder circuits were successfully implemented and tested. The outputs matched expected logic behavior. Minor wiring issues were resolved with careful checking.

The experiment confirmed the correct functionality of all circuits. These components are essential in digital systems for data routing and signal control. Overall, the objectives were achieved effectively.

**Reference:**

1. Thomas L. Floyd, “Digital Fundamentals” 9th edition, Prentice Hall.
2. M. Morris Mano, “Digital Logic & Computer Design” Prentice Hall.