North South University

Department of Electrical & Computer Engineering

Lab Report

Experiment No:	**01**
Experiment Title:	**Design of a 2-bit Logic unit**
Course Code:	**CSE332L**
Course Name:	**Computer Organization & Architecture Lab**
Name & ID:	**Ishrat Jahan,1921909042**
Date of Experiment:	**28/10/2021**
Date of Submission:	**29/10/2021**

Objective:

1: To construct a 2-bit logic unit which will perform logic operations AND, OR, XOR and NOT

2: Labelling the inputs, outputs and correct selection of the multiplexer.

3: Adjusting the inputs and output's data bits and the selection bits of MUX, for proper functioning of a 2-bit logic unit.

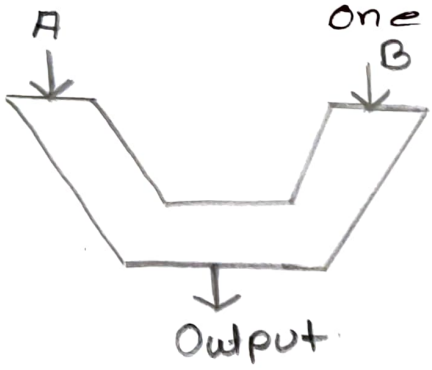
Theory:

ALU: ALU stands for arithmetic and logic unit. It's a part of microprocessor and performs two kinds of operations: Arithmetic and Logic.

Arithmetic operations includes addition, subtraction, multiplication and division.

Logic operations includes AND, OR, XOR, XNOR and NOT. These operations can manipulate bits, change values of bits, delete or insert new bits in a register.

2-bit logic unit: A 2-bit logic unit will have two inputs, each of them being 2 bits. Therefore we will have two outputs, one output for each of the 2 bits.

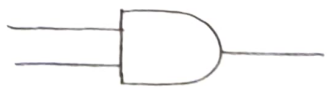


$\therefore A = A_1 A_0$

$\therefore B = B_1 B_0$.

IC: 7404: NOT gate, 7408: AND gate, 7432: OR gate
7486: XOR gate, 74f153: Dual 4:1 MUX.

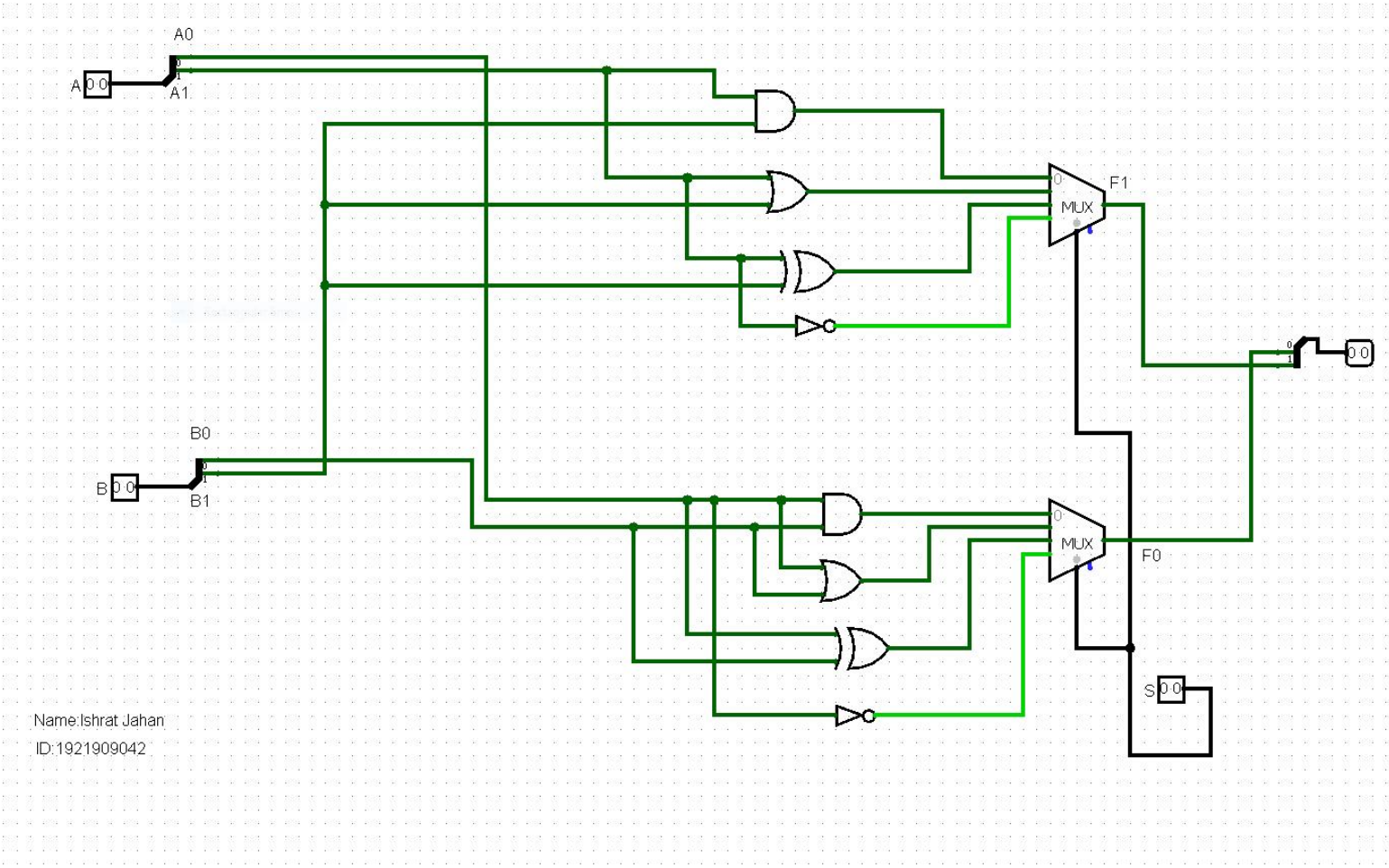
Logic Gates:


[AND gate]


[OR gate]


[XOR gate]


[NOT gate].


A0
A
0 0
A1
B0
B
0 0
B1
MUX
F1
MUX
F0
0 0
S
0 0
Name:Ishrat Jahan
ID:1921909042

Truth Table

A1	A0	B1	B0	AND1	AND0	OR1	OR0	XOR1	XOR0	NOTA1	NOT A0
0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	1	0	1	1	1
0	0	1	0	0	0	1	0	1	0	1	1
0	0	1	1	0	0	1	1	1	1	1	1
0	1	0	0	0	0	0	1	0	1	1	0
0	1	0	1	0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1	1	1	1	0
0	1	1	1	0	1	1	1	1	0	1	0
1	0	0	0	0	0	1	0	1	0	0	1
1	0	0	1	0	0	1	1	1	1	0	1
1	0	1	0	1	0	1	0	0	0	0	1
1	0	1	1	1	0	1	1	0	1	0	1
1	1	0	0	0	0	1	1	1	1	0	0
1	1	0	1	0	1	1	1	1	0	0	0
1	1	1	0	1	0	1	1	0	1	0	0
1	1	1	1	1	1	1	1	0	0	0	0

Discussion

The experiment of Lab01 was based on "Design of a 2-bit Logic unit" which is a part of an ALU. In this lab, we learned how to design a 2-bit logic unit in logisim.

At first, we studied the concept of logic unit in ALU. We learnt that ALU performs arithmetic and logical operations and how it can be used to perform several operations like insertion of bits, changing bits, manipulating or deleting bits. For example: If a register has 1010 stored in it and we are asked to clear the data, we can either perform an AND operation with 0000 or perform XOR operation of the number with itself. It will result in 0000. This is called manipulation. We also studied what each IC stands for.

Secondly, we proceeded to learn what 2 bits mean in a logic unit. For a 2-bit logic unit, we need two inputs A and B each of them being 2 bits; $A = A_1A_0$, $B = B_1B_0$ (where A_0 and B_0 is the least significant bit and A_1 and B_1 is the most significant bit). As we were asked to perform 4 logical operations with the logic unit (XOR, AND, OR, NOT), we chose equipments according to them. Following this, we used our theoretical knowledge to complete the truth-table.

Then we proceeded to design the 2-bit logic unit in Logisim. At first, I constructed the two 2-bits input. For that I took a input pin and changed it's data bits to 2 and labelled it as 'A'. I repeated the same process for the second input 'B'. Then I took two splitter from the wiring section and connected it to the both inputs. The 0 part in the splitter stands for A_0 and B_0 whereas the 1 part is for A_1 and B_1. Therefore I labelled them accordingly. As the input's are 2 bits, I adjusted the fan out and bit width of splitter to 2. Also I learnt how I can change the facing of the splitter and how to adjust the position of 0 and 1 on it.

Secondly, I took two MUX from the plexers section and changed the select bits to 2 which results in 4:1 mux. As we will be performing one logical operation at a time and be deciding which logical operation the circuit will be performing, we used MUX so we can control it using selection bits. As the logic unit is 2 bits and we had to perform four operations, I used 4:1 MUX. If we had five operations, I would have used 8:1 MUX.

For adjusting the selection bits, I took an input pin, changed it's data bit to 2, and connected it to both of the MUX's selection bit port. According to the diagram provided, 00 is for AND operation, 01 for OR, 10 for XOR and 11 for NOT.

After that, I took AND gate, OR gate, NOT gate and XOR gate from the gates section and connected them with both multiplexers, each of them having individual of all of these four gates. As per the truth table, I needed to perform operation between A_1 and B_1 and A_0 and B_0. So I connected, the inputs in this order with the logic gates. The first MUX carried out the operation between. A_1 and B_1 and the second one between A_0 and B_0. We avoided any kind of shocks while connecting them.

Lastly, for displaying the output, ~~we~~ I chose the output pin and changed it's data bit to 2. The shape of this output pin is more like a rounded rectangle and we can easily distinguish it from the input pin which has a sharp shape. After that, I connected the output pin with a splitter. The output from $A_0 B_0$ MUX was labelled as F_0 and connected to splitter's 0 part and $A_1 B_1$ was labelled as F_1 and connected to splitter's 1 part. The whole circuit was completed.

Finally, after completion, I checked the outputs and matched them with my truth table. The circuit worked correctly and the experiment was successful.

The only limitation in this experiment was, it was supposed to be a hardware based but we constructed it in a software. Therefore, our practical learning was limited to an extent. As we are not practicing them with equipment's, it might result in not getting a good hand in building complex circuits in future. Overall, if we could have conducted it in a lab, it could have been better for us.