Digital Electronics & Microprocessors Laboratory (EC2P006)

EXPERIMENT-2

Shorya Sharma
19EE01017

Part 1

Aim:

Install and simulate the basic logic circuit in Logisim; and then reporting the results along with the logic circuits.

We need to simulate the expression y = A.B + B.C’ + C.A’ and then find out its corresponding truth table.

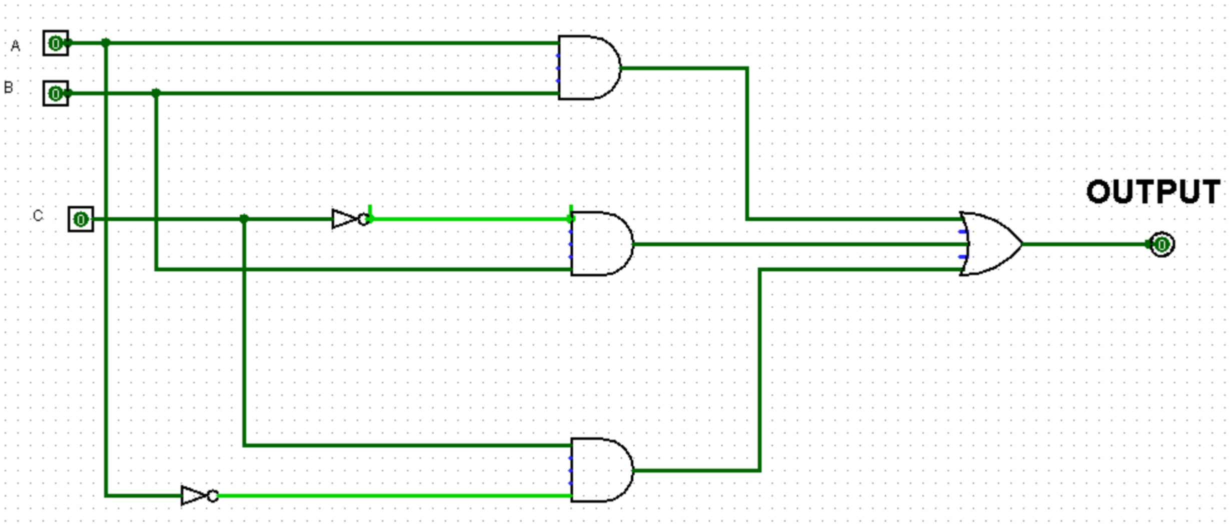
Theory:

And Gate: The AND gate is a basic digital logic gate that implements logical conjunction - it behaves according to the truth table to the right. A HIGH output (1) results only if all the inputs to the AND gate are HIGH (1). If none or not all inputs to the AND gate are HIGH, LOW output results.

Or Gate: The OR gate is a digital logic gate that implements logical disjunction – it behaves according to the truth table to the right. A HIGH output (1) results if one or both the inputs to the gate are HIGH (1). If neither input is high, a LOW output (0) results. In another sense, the function of OR effectively finds the maximum between two binary digits, just as the complementary AND function finds the minimum.

Not Gate: In digital logic, an inverter or NOT gate is a logic gate which implements logical negation.

Circuit Diagram:



Truth Table

A	B	C	Output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Discussion

The above logic circuit implemented using AND, OR and NOT gates essentially builds up a logic 3-bit circuit that operates on 3 inputs A, B and C and gives the desired truth table as per the equation y(Output)=A. B + B.C’ + C.A’

Part 2(All basic Logic gates using NAND and NOR)

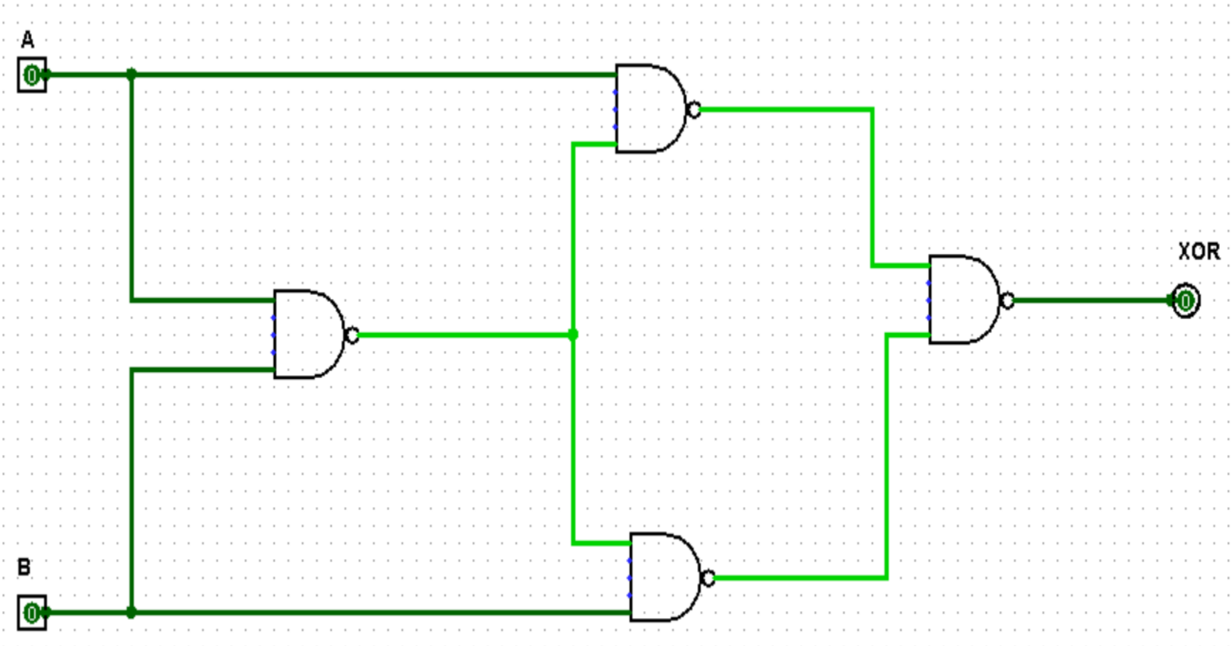
Aim:

To form a XOR logic gate using multiple NAND gates.

Theory:

A Boolean operator which gives the value zero if and only if all the operands have a value of one, and otherwise has a value of one (equivalent to NOT AND).

Circuit Diagram:



Truth Table:

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

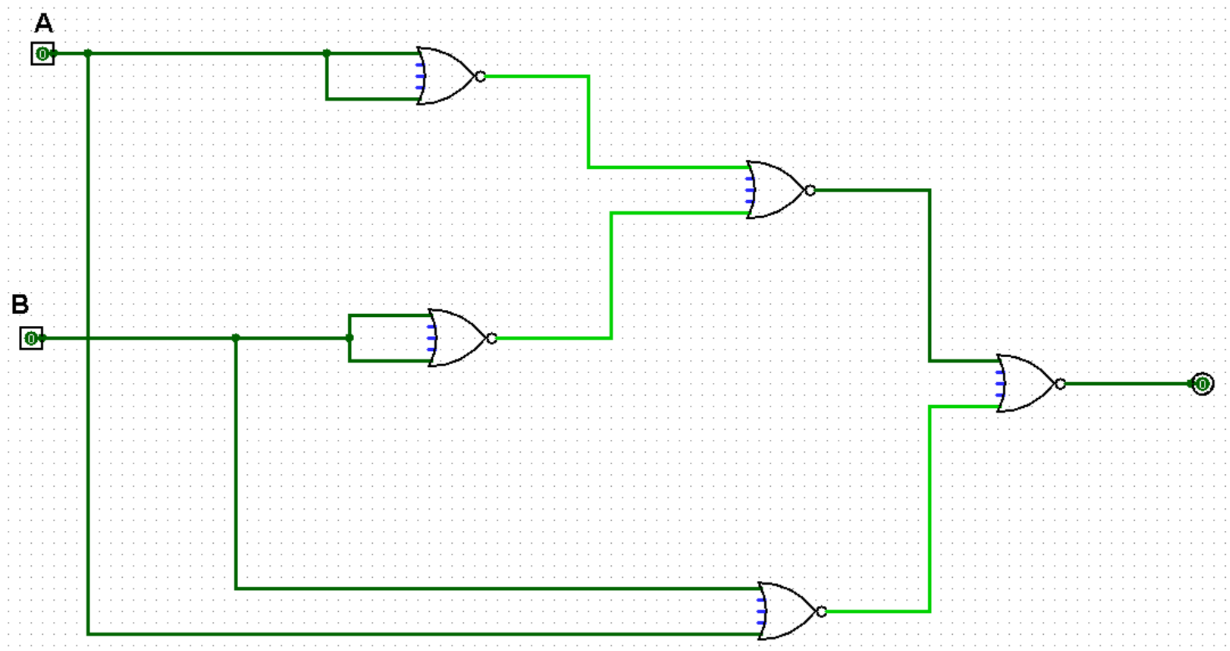
Aim:

To form a XOR gate using multiple NOR Gates

Theory:

The NOR gate is a digital logic gate that implements logical NOR - it behaves according to the truth table to the right. A HIGH output results if both the inputs to the gate are LOW; if one or both input is HIGH, a LOW output results. NOR is the result of the negation of the OR operator.

Circuit Diagram:



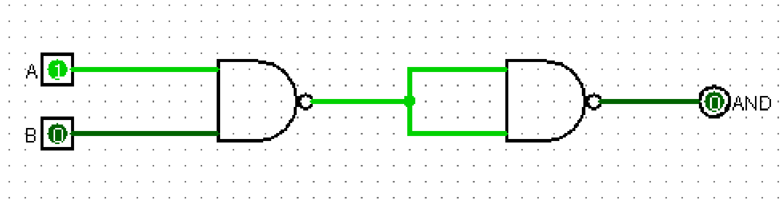
Truth Table:

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Aim:

To form AND Gate using NAND Gate.

Circuit Diagram:



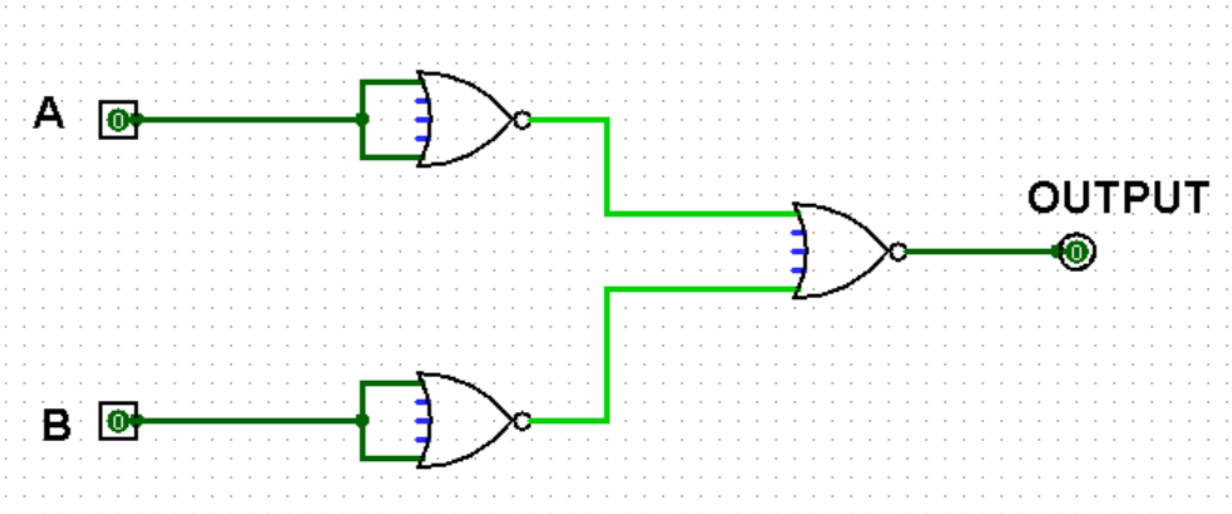
Truth Table:

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

Aim:

To form AND Gate using NOR Gate

Circuit Diagram:



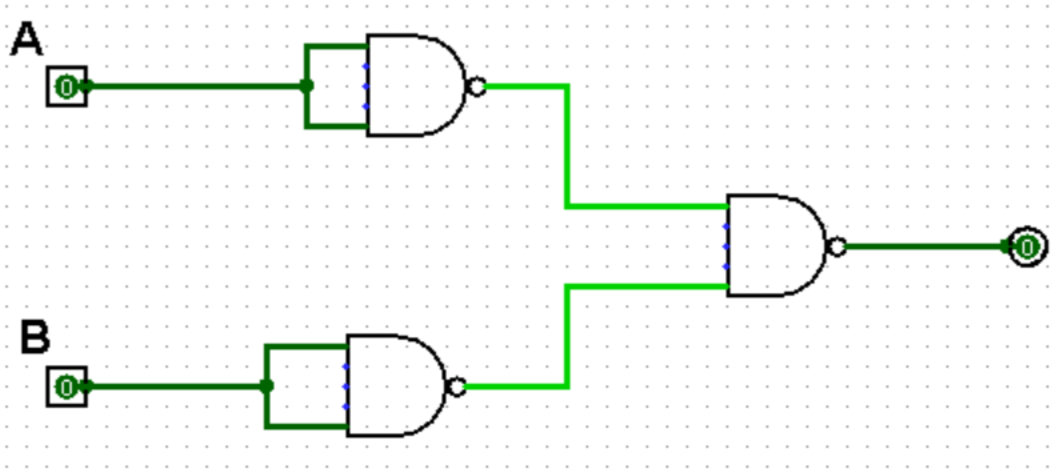
Truth Table:

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

Aim:

To form OR Gate using NAND Gate

Circuit Diagram:



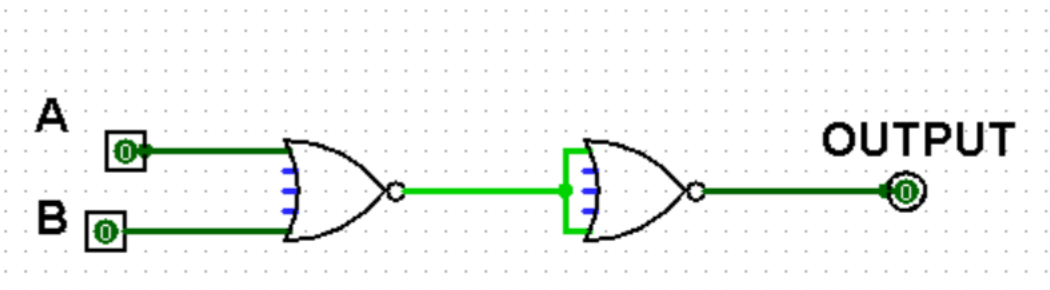
Truth Table:

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

Aim:

To form OR Gate using NOR Gate

Circuit Diagram:

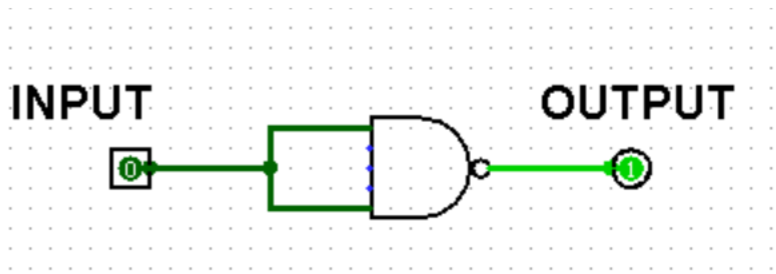


<u>Truth Table:</u>

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

<u>Aim</u>:
To form NOT Gate using NAND Gate

<u>Circuit Diagram</u>:

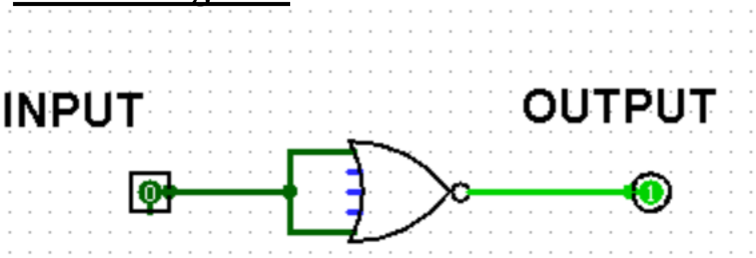


<u>Truth Table</u>:

A	NOT
0	1
1	0

Aim:
To form NOT Gate using NOR Gate

Circuit Diagram:



Truth Table:

A	NOT
0	1
1	0

Discussion

Thus, in Part 2, we designed all gates-XOR, AND, OR, NOT using the Universal Gates of NAND and NOR; all such gates form an important part of Digital Logic Circuits and Microprocessors.

Part 3

Aim:

Simulation and Implementation of Half Adder.

Theory:

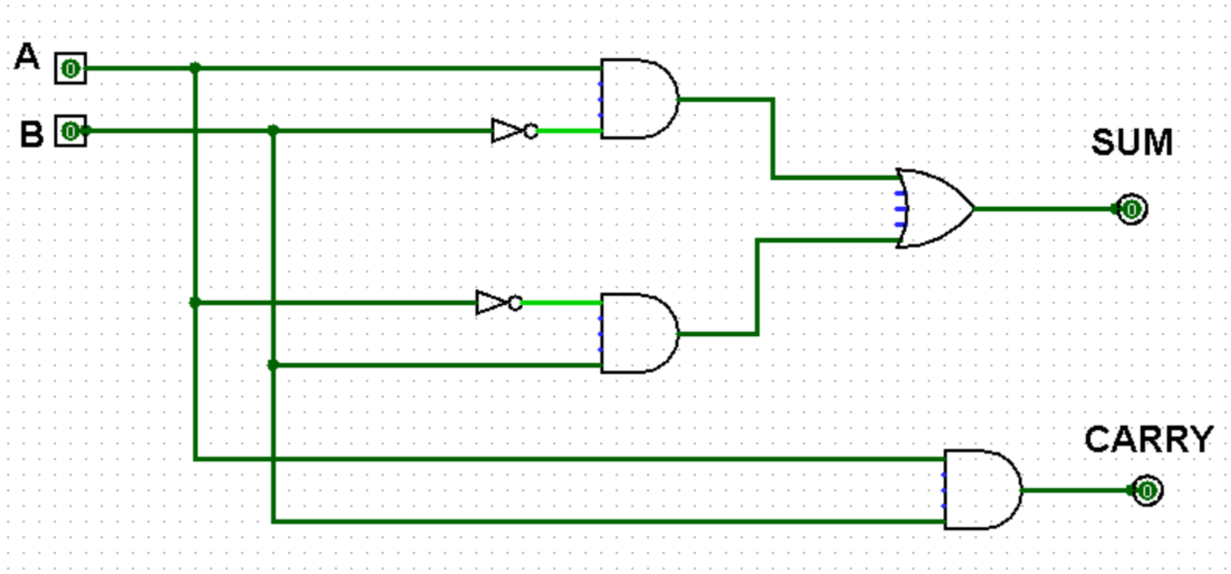
The addition of 2 bits is done using a combination circuit called Half adder. The input variables are augend and addend bits and output variables are sum & carry bits. A and B are the two input bits.

Half Adder

Truth Table:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Circuit Diagram:



Truth Table:

A	B	S	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Discussion

For designing the Half Adder, I used NAND Gates to design an XOR Gate for the S part and an AND Gate for the Carry over Cout part.

Conclusion

Logic gates perform basic logical functions and are the fundamental building blocks of digital integrated circuits. Most logic gates take an input of two binary values, and output a single value of a 1 or 0.
Some circuits may have only a few logic gates, while others, such as microprocessors, may have millions of them.

----------X---------