Computer Circuit Fundamentals (Lab 5)

More complex experiments with Logic Gates

Leonardo Fusser, 1946995

Experiment Performed on **4 October 2019**
Report Submitted on **18 October 2019**

Department of Computer Engineering Technology
Computer Circuit Fundamentals
Subash Handa

VANIER
CÉGEP / COLLEGE
Learning today Leading tomorrow

Table of Contents

OBJECTIVES

- Use NAND gates to implement AND, OR and NOT functions.
- Use NOR gates to implement AND, OR and NOT functions.
- To further understand Boolean logic/algebra.
- To further enhance understanding with ICs.
- To further enhance understanding with logic gates.

DESIGN

Part 1 Experiment

- ***Part 1*** experiment consisted of logic implementation of basic gates with NAND or NOR gates. Part 1a consisted of implementing AND, OR and NOT gates *using only* **NAND gates** (refer to Question 1a logic diagrams under "Schematics" for reference and for logic circuit diagrams and truth tables). Part 1b consisted of implementing AND, OR and NOT gates *using only* **NOR gates** (refer to Question 1b logic diagrams under "Schematics" for reference and for logic circuit diagrams and truth tables).

Part 2 Experiment

- ***Part 2*** experiment consisted of logic implementation of a Boolean expression **using NAND gates**. This Boolean expression *(F = AB'C' + A'BC' + A'B'C + ABC)* was simplified to *(A + B + C)* which is similar to an **XOR expression**. With this simplified expression, a logic circuit was created *using only* **NAND gates** and that circuit was tested with the simplified expression (Refer to Question 2 logic diagram under "Schematics" for reference and for logic circuit diagram and truth table).

Part 3 Experiment

- ***Part 3*** experiment consisted of logic implementation of a Boolean expression **using NOR gates**. This Boolean expression *(F = AB'C' + A'BC' + A'B'C + ABC)* was simplified to *(A + B + C)* which is similar to an **XOR expression**. With this simplified expression, a logic circuit was created *using only* **NOR gates** and that circuit was tested with the simplified expression (Refer to Question 3 logic diagram under "Schematics" for reference and for logic circuit diagram and truth table).

Part 4 Experiment

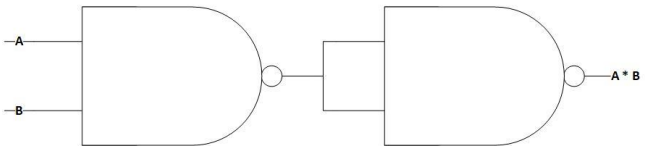
- ***Part 4*** experiment consisted of creating a logic circuit diagram of a 3-input majority gate (otherwise known as a 3-input AND gate) *using only* **NAND gates**. The circuit was tested based on the logic of an AND gate. (Refer to Question 4 logic diagram under "Schematics" for reference and for logic circuit diagram and truth table).

SCHEMATICS

Question 1a logic diagram

Simulating an AND gate using a NAND gate

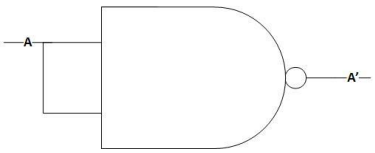
Input		Output
A	B	F = A.B
0	0	0
0	1	0
1	0	0
1	1	1



Question 1a logic diagram

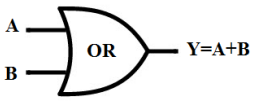
Simulating a NOT (inverter) gate using a NAND gate

A	**NOT** A
0	1
1	0

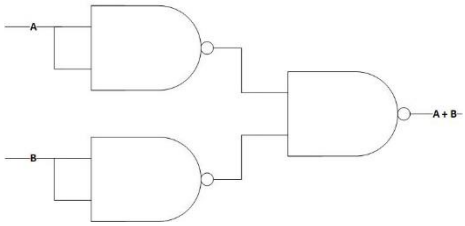


Question 1a logic diagram

Simulating an OR gate using a NAND gate



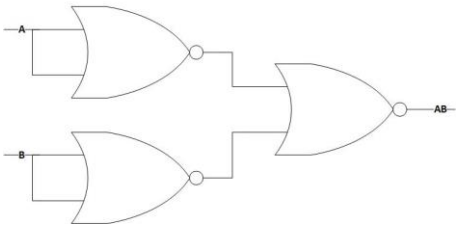
Inputs		Output
A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1



Question 1b logic diagram

Simulating an AND gate using a NOR gate

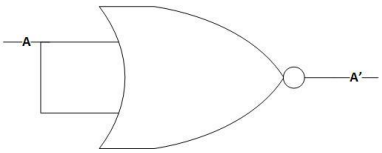
Input		Output
A	B	F = A.B
0	0	0
0	1	0
1	0	0
1	1	1



Question 1b logic diagram

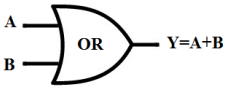
Simulating a NOT (inverter) gate using a NOR gate

A	**NOT** A
0	1
1	0

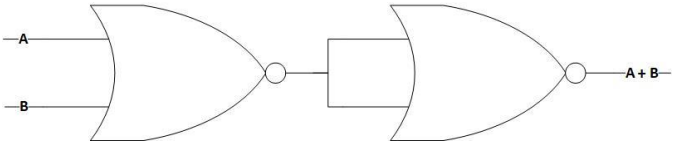


Question 1b logic diagram

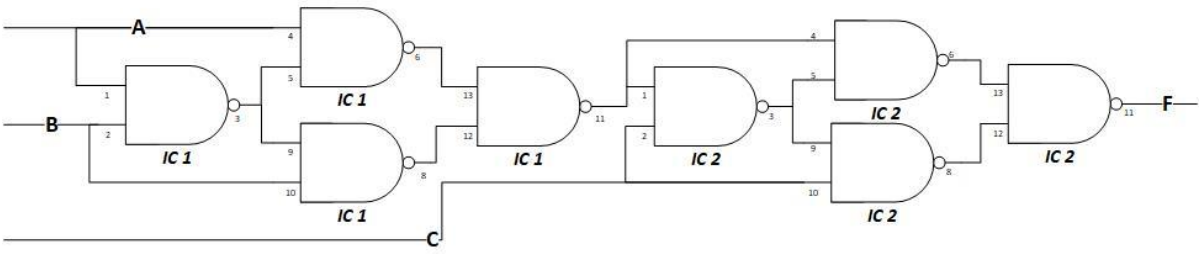
Simulating an OR gate using a NOR gate



Inputs		Output
A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1



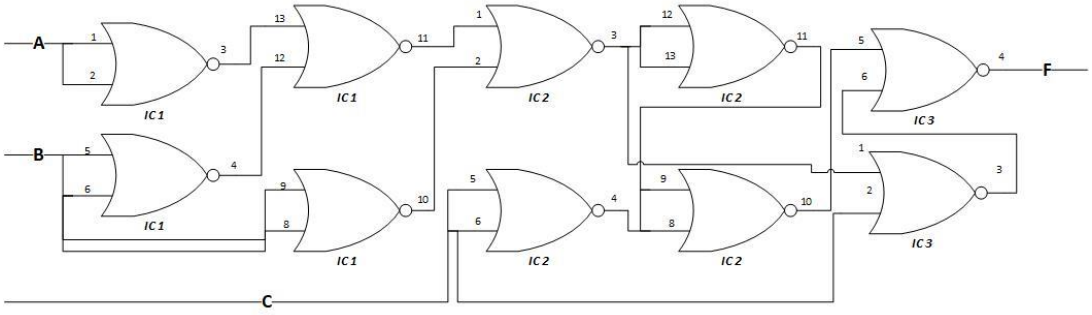
Question 2 Logic Diagram



3-input XOR gate

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

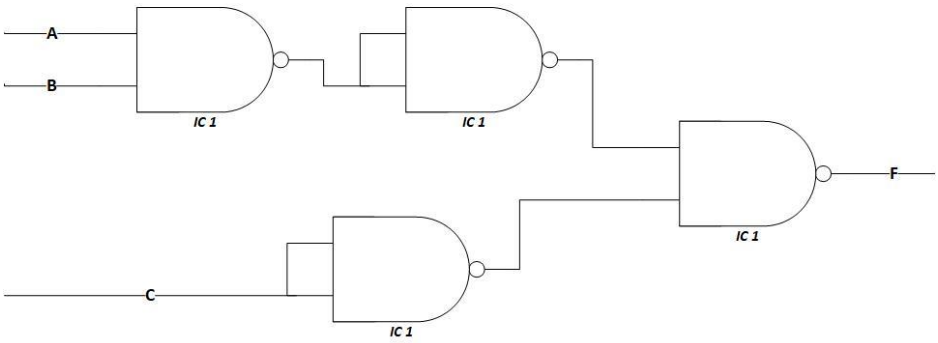
Question 3 Logic Diagram



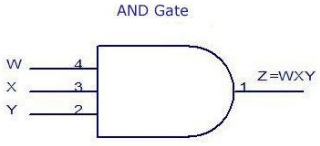
3-input XOR gate

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

Question 4 Logic Diagram



3 Input AND Gate



TRUTH TABLE

INPUTS			OUTPUT
W	X	Y	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

QUESTIONS

For Part 1

A: AND, OR and NOT gates were implemented only using NAND gates. A logic circuit diagram was created along with a truth table proving the logic (refer to Part 1 experiment under "Design" for reference).

B: AND, OR and NOT gates were implemented only using NOR gates. A logic circuit diagram was created along with a truth table proving the logic (refer to Part 1 experiment under "Design" for reference).

For Part 2, Part 3, Part 4

A: All circuits were wired and logic diagrams were developed based on those circuits.

B: All circuits were tested and the logic was debugged if any issues were present.

C: The instructor verified the logic for all circuits.

(refer to Part 2, Part 3 and Part 4 experiment under "Design" for reference)