Lab Work 2 (Part 1)

Implementing Combinational Logic

There are a few skills that are essential in digital electronics design:

- building logic circuits from schematics
- identifying the boolean equation for a logic circuit (schematic)
- simplifying a logic design (by simplifying boolean equation - algebra OR K-Map)
- drawing a logic circuit schematic from a boolean equation

If time permits, we will also look at the NAND/NOR gates as *universal* gates.

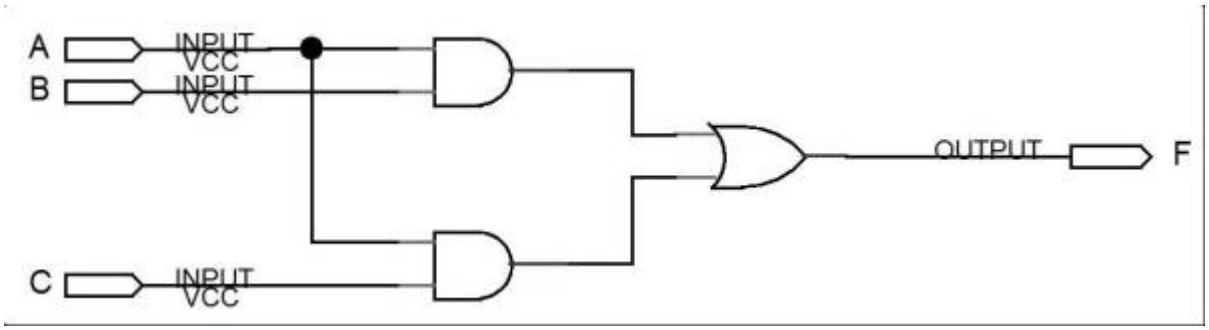
Note: After a while, you will realize that most (if not all) logic circuits can be implemented using AND-OR logic (a group of AND gates followed by a group of OR gates) implementation. In Boolean algebra, the equation for this implementation is in Sum-of-Product (SOP) form.

From Schematics to Breadboards

This is simply an extension to what have been done in previous lab sessions.

Exercise 2.1

Build the circuit shown below (Figure 2.1) on a breadboard:

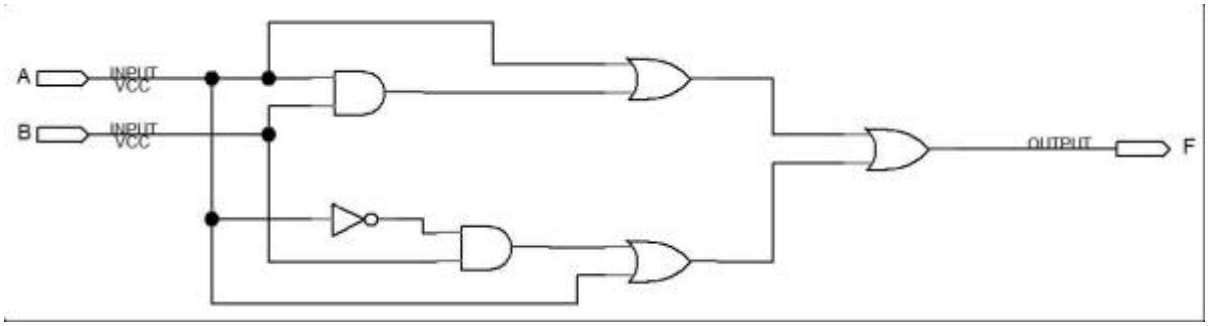


Build a truth table for the circuit.

Notice that the circuit is a direct implementation of SOP form boolean equation.

Exercise 2.2

Build the circuit shown below (Figure 2.2) on a breadboard:

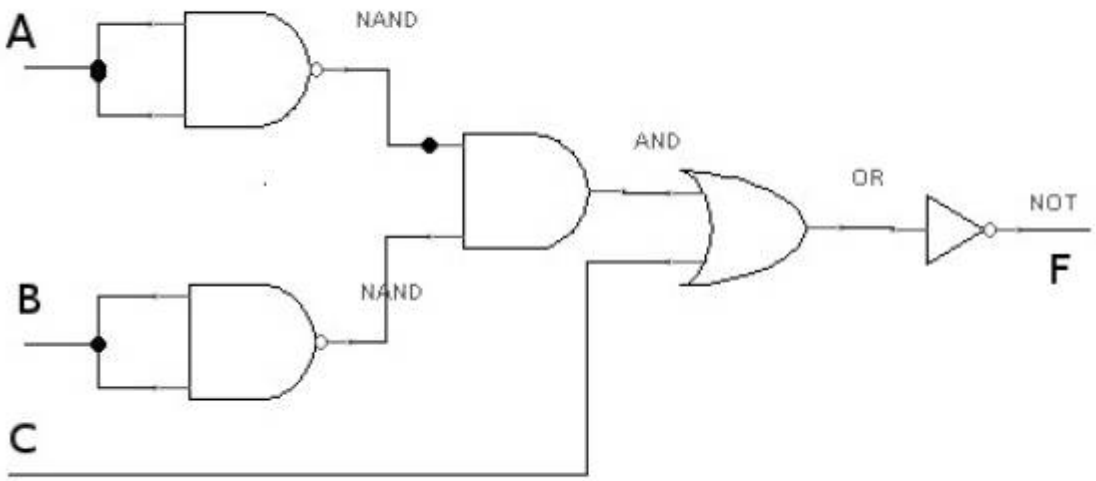


Build a truth table for the circuit.

Notice that the circuit is also a direct implementation of SOP form boolean equation.

Exercise 2.3

Build the circuit shown below (Figure 2.3) on a breadboard:



Build a truth table for the circuit.

From Schematics to Equations

Work related to this can be done on paper without having to have any hardware. However, some hardware verification is always desirable since that is what this is all about (implementing logic on hardware).

Exercise 2.4

Derive Boolean equations for the circuits given in Figure 2.1 and Figure 2.2. You should get an equation in SOP form. For each equation,

- verify the equation using the truth table generated earlier
- rewrite the equation into a standard SOP form
- verify the 'extended' equation

Exercise 2.5

Derive Boolean equation for the circuit given in Figure 2.3.

- verify the equation using the truth table generated earlier
- try to redesign the circuit (draw schematic) so that it uses least number of ICs
- construct the new circuit and verify the design

From Equations to Schematics

The main task here is to identify the logic gates needed from a given Boolean equation.

Exercise 2.6

Consider this Boolean equation: $Y=AB+AC+A$

Simplify the equation. Build the circuit and construct a truth table.

***Note**: You can practice your knowledge by constructing standard SOP form of the given equation*

Exercise 2.7

Consider this Boolean equation: $Y=AB+BC(B+A)$

Simplify the equation. Build the circuit and construct a truth table.

***Note**: You can practice your knowledge by constructing standard SOP form of the given equation*

Exercise 2.8

Consider this truth table:

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Find the Boolean equation that satisfy the given truth table. Build the circuit and verify.

***Note**: You can practice your knowledge by trying to simplify the equation using K-Map*

Exercise 2.9

Consider this truth table:

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

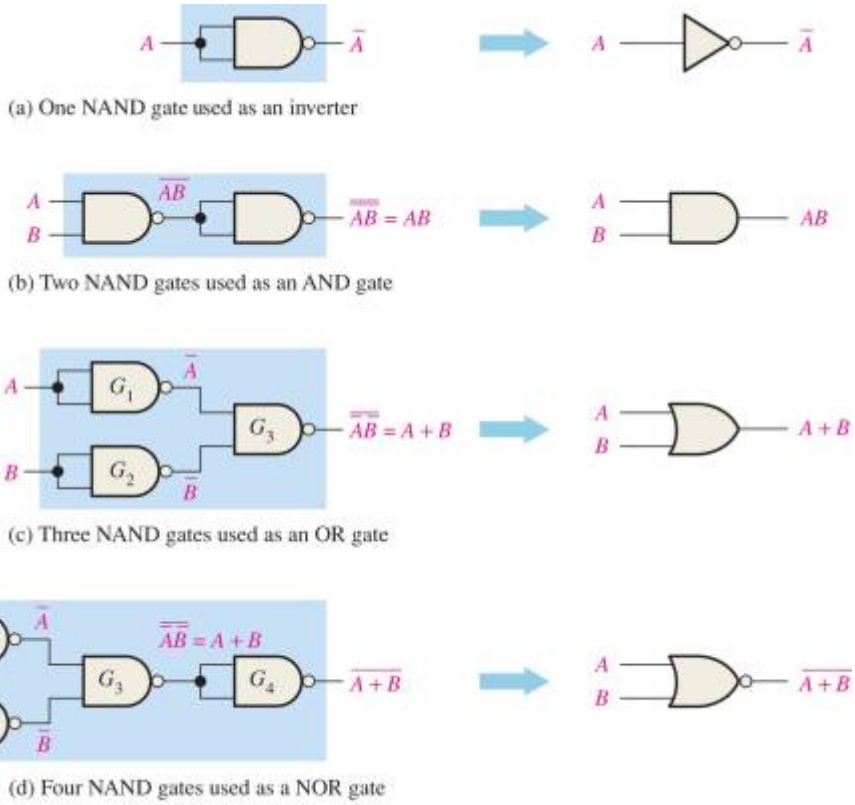
Find the Boolean equation that satisfy the given truth table. Build the circuit and verify.

***Note**: You can practice your knowledge by trying to simplify the equation using K-Map*

Universal Gates

The NAND and NOR gates are also known as universal gates due to the fact that they can be used to emulate other logic gates.

Exercise 2.10



(a) One NAND gate used as an inverter

(b) Two NAND gates used as an AND gate

(c) Three NAND gates used as an OR gate

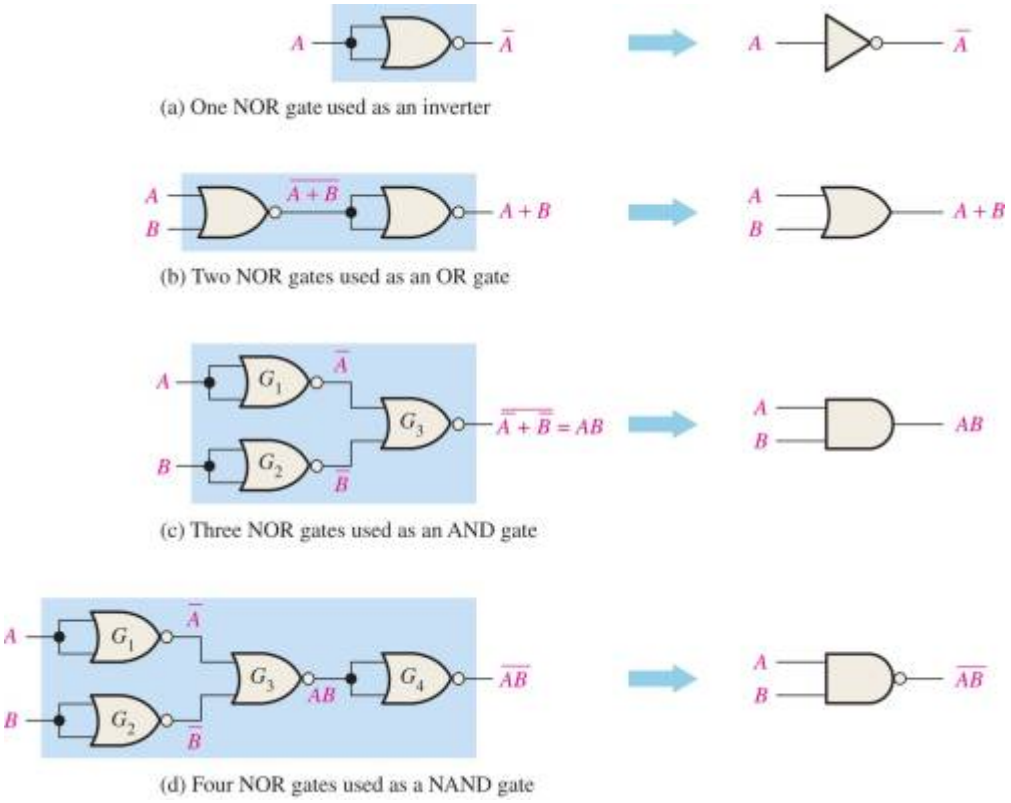
(d) Four NAND gates used as a NOR gate

***Disclaimer**: This image is extracted from resources available for Digital Fundamentals 11th Edition (Global Edition)*

Use NAND gates to implement these equations:

- $X=\overline{A}+B$
- $X=A\ \overline{B}$

Exercise 2.11



Disclaimer: *This image is extracted from resources available for Digital Fundamentals 11th Edition (Global Edition)*

Use NOR gates to implement these equations:

- $X=\overline{A}+B$
- $X=A\ \overline{B}$

From:
https://azman.my1matrix.cc/ - **Azman @MY1**

Permanent link:
https://azman.my1matrix.cc/doku.php?id=archive:pgt104lab01a

Last update: **2026/02/08 03:46**