EXPERIMENT 5

Basic Digital Logic Circuits

Introduction

The experiments in this laboratory exercise will provide an introduction to digital electronic circuits. You will learn how to use the Bit Bucket breadboarding system to build circuits using common logic gates including NAND's, NOR's and inverters.

Experiment Objectives:

- Learn basic principles of digital logic
- Learn how to use the Bit Bucket breadboarding system
- Continue to develop professional lab skills and written communication skills.

Bring to Lab:

Your completed Pre-Lab. Turn this in when you get to lab.
Several sheets of Engineering Paper.

Theory: Digital Logic (Binary)

Almost all computers today use binary digital logic circuits. These circuits have just two possible output voltages, which can be called by any contrasting terms; the most common are "HIGH / LOW", or "TRUE / FALSE", or "ONE /.ZERO." Such an output is called a "binary digit," or *bit*.

The decimal numbers and alphabet characters we are familiar with are converted to binary bits before they are fed into a computer's arithmetic logic unit (ALU). Inside the ALU, the computer executes a program and generates binary results. The binary results are converted back to decimal numbers, alphabet letters, graphics, or sound so we can understand them.

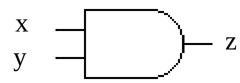
Most of the fundamental data processing inside computers is done using *logic gates*. Logic gates combine individual bits according to certain rules. These rules, taken together, form the basis of *Boolean algebra*, which you may study in depth in various lecture courses.

Logic Gates

We will introduce the most common logic gates in this section, including the AND, OR, XOR, NOT, NOR, and NAND. For each gate, we will show the circuit symbol, the Boolean algebra logic function, and the truth table. The truth table lists all possible combinations of inputs, and the resulting output for each.

AND Gate
The AND gate can have any number of inputs. The output is HIGH (1) only if all the inputs are HIGH.

LOGIC SYMBOL:



2-input AND gate

BOOLEAN ALGEBRA EQUATION: $z = (x \text{ AND } y) = (x \bullet y) = xy$

TRUTH TABLE:

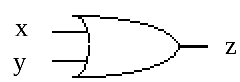
Table 1. Truth table for a 2-input AND gate.

x	y	$z = x \bullet y$
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

An OR gate can have any number of inputs. The output is HIGH (1) if one or more of the inputs is HIGH.

LOGIC SYMBOL:



2-input OR gate

BOOLEAN ALGEBRA EQUATION: $z = (x \text{ OR } y) = (x + y)$

TRUTH TABLE:

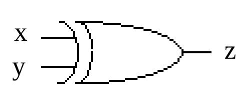
Table 2. Truth table for a 2-input OR gate.

x	y	$z = x + y$
0	0	0
0	1	1
1	0	1
1	1	1

Exclusive-OR Gate (XOR)

A two-input XOR gate is shown below. The output is HIGH (1) if exactly one of the inputs is HIGH.

LOGIC SYMBOL:



2-input XOR gate

BOOLEAN ALGEBRA EQUATION: $z = (x \text{ XOR } y) = (x \oplus y)$

TRUTH TABLE:

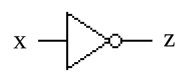
Table 3. Truth table for a 2-input XOR gate.

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

INVERTER (NOT Gate)

An inverter has just one input. The output is the *logical complement* of the input. If the input is HIGH (1), the output is LOW (0), and vice-versa. In Boolean algebra notation, the complement operation is denoted with an over-bar as shown in the equation. On circuit diagrams, it is indicated with an open circle, as shown in the circuit diagram.

LOGIC SYMBOL:



Inverter (NOT gate)

BOOLEAN ALGEBRA EQUATION: $z = \text{NOT } x = \overline{x}$

TRUTH TABLE:

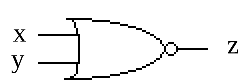
Table 4. Truth table for an inverter.

x	$z = \overline{x}$
0	1
1	0

NOR Gate

A NOR (NOT-OR) gate is the combination of an OR and a NOT. The output is LOW (1) if one or more of the inputs is HIGH.

LOGIC SYMBOL:



2-input NOR gate

BOOLEAN ALGEBRA EQUATION: $z = (x \text{ NOR } y) = \overline{x + y}$

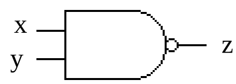
TRUTH TABLE:

Table 5. Truth table for a 2-input NOR gate.

x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

NAND Gate
A NAND gate is the logical combination of an AND and a NOT. The output is LOW (1) if all of the inputs are HIGH.

LOGIC SYMBOL:



2-input NAND gate

BOOLEAN ALGEBRA EQUATION: $z = (x \text{ NAND } y) = \overline{x \bullet y} = \overline{xy}$

TRUTH TABLE:

Table 5. Truth table for a 2-input NAND gate.

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

Inputs and Outputs
For studying digital logic circuits we will use several types of inputs and outputs, including switches, clock signals, LED’s, and 7-segment displays.

Switches
A switch is used to manually connect the input of a gate to a HIGH or LOW voltage. We will use pushbutton (PB) and toggle (T) switches.

Clock
A Clock is a circuit that produces a periodic output that alternates between HIGH and LOW voltage. We will use a clock signal to test some of our logic circuits.

LED’s and 7-segment displays
Light-Emitting Diodes (LED’s) can be used to display the output of a logic gate. Usually, they are connected so that if the output is HIGH, the LED is on. Seven--segment displays are packaged arrays of LED’s that are used to display numbers and letters.

Logic Families

A logic family is a complete set of logic gates that are manufactured using a particular type of electronic circuitry. There are numerous commercially available logic families to suit different design requirements. The most common logic families are listed in the table below, together with their relative advantages and disadvantages.

Table 6. Some common logic families

Acronym	Full name	Advantages	Disadvantages
CMOS	Complementary metal-oxide-semiconductor	Lowest power consumption. Most common logic family- used in all microcomputer chips today.	Easily damaged by static discharge and voltage spikes.
TTL	Transistor-transistor logic	Earliest developed. Most rugged – least susceptible to electrical damage.	Consumes more power than CMOS – not suitable for battery operated devices.
ECL	Emitter-coupled logic	Fastest available logic family	Consumes more power than CMOS. Requires extreme care in wiring.

TTL Logic

In this experiment, we will be using TTL logic. All chips in the TTL logic family have the following specifications:

- POWER SUPPLIES: +5 V and Ground (0 V).
- LOGIC HIGH: 2.0 to 5 V
- LOGIC LOW: 0 to 0.8 V
- FAN-OUT (Number of inputs that each output can be connected to): 10

More detail can be found in any textbook on digital electronics, and also on the internet. Here is one useful website for TTL information: http://www.twysted-pair.com/74xx.htm

Bit-Bucket Digital Logic Breadboard System

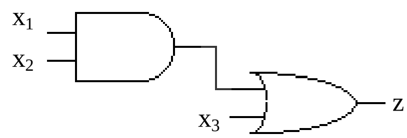
In this experiment, we will use a breadboard system especially designed for TTL digital logic. It provides regulated 5 V power, switches for manual input, and an adjustable clock. It also provides LED's and 7-segment displays for monitoring outputs.

Circuit Examples

Here are some examples of digital logic circuits. For each example, either the circuit diagram, Boolean expression, or truth table are given and the problem is to find the other two representations. You should be able to follow these examples to work the pre-lab questions.

Example 1.

Given the logic circuit below, find the Boolean expression and write the truth table.



Solution: This circuit includes an AND gate with inputs x_1 and x_2, the output of which is OR'ed with x_3. The Boolean expression is written by inspection:

$$z = x_3 + (x_1 \bullet x_2)$$

The truth table is written by listing every possible combination of the inputs, and calculating the output for each one with the help of the basic truth tables listed previously for AND and OR. As shown below, it is helpful to include intermediate results in the table. In this case, an extra column is inserted for the AND gate output, $x_1 \bullet x_2$.

x_1	x_2	x_3	$x_1 \bullet x_2$	$z = x_3 + (x_1 \bullet x_2)$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

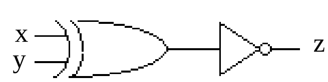
Example 2

Given the following truth table, design the circuit and write the Boolean equation:

x	y	z
0	0	1
0	1	0
1	0	0
1	1	1

Solution: We observe that the output, z, is the complement of the XOR function. Therefore we can implement this as an XOR gate followed by an inverter.

$$z = \overline{x \oplus y}$$



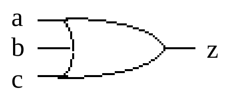
(This function actually has a name of its own – it is called the XNOR. The circuit symbol is an XOR gate with a complement circle on the output.)

Example 3

Given the following Boolean expression, design the circuit and write the truth table:

$$z = a + b + c$$

Solution. We recognize this as a three-input OR function. The circuit is drawn below:



The output is HIGH if any of the inputs are HIGH. The truth table is:

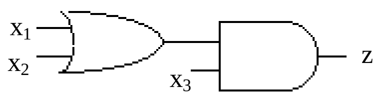
a	b	c	$z = a + b + c$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Your Name____

Prelab Questions (10 points)

Answer these questions before coming to lab and turn them in when you arrive. You may do your work on separate paper (for example you might want to do your work on a computer), but please attach your work to this sheet for submission.

1. Write the Boolean expression and the truth table for this logic circuit:



2. Design the circuit and write the Boolean expression for the given truth table.

a	b	c	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

3. Go to the TTL web site referred to in the pre-lab and use the given information to define the term “Noise Margin” in your own words.

4. A logic circuit is given below with the inputs applied as shown. Label each logic gate, and determine the value of the output z (1 or 0).