Lab Assignment 3 – Logic Gates

Laboratory Goal:	To use your existing knowledge of voltage concepts to design simple logic circuits.
Learning Objectives:	Operation of simple logic gates
Suggested Tools:	Logic integrated circuits (ICs), LEDs, datasheets, breadboard, multimeter.

Background

In this lab, we'll learn about a class of circuit elements called *logic gates* that are capable of measuring voltages and making decisions based on those measurements. Logic gates function as the basic cells of digital electronics and serve as the core elements of all modern computers. Using combinations of different logic gates, complex decision-making operations can be performed.

A logic gate is fundamentally a binary device which interprets the voltage values 0 V and 5 V as representing the binary (i.e., base-two) digits **0** and **1**. In principle, any voltage in a logic system that is smaller than the midpoint value of 2.5 V can be interpreted as a logic **0**, and anything above 2.5 V as a logic **1**. In a well-designed logic system, however, a given voltage is held as close as possible to one of the defined values of 0 V and 5 V, except, of course, in transition from one value to another.[1]

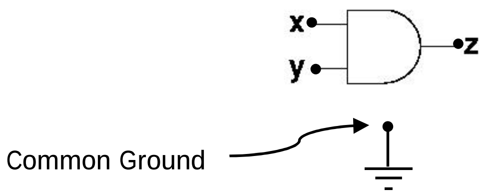
The typical logic gate has two or more input terminals and a single output terminal. Terminal voltages are defined and measured relative to a common ground terminal which is often omitted in circuit diagrams for clarity. A common ground connection is understood to always be present. In order to function, a logic gate must also be energized by connecting separate power and ground terminals to a 5-V voltage source. For historical reasons, the power terminal of a logic gate is often labeled V_{CC} on its data sheet.

A *truth table* is often used to describe the way in which a given logic gate functions. The table contains the complete set of all possible binary input combinations and their resulting output values produces by the gate in response. The relationship between the input and output values depends of the specific type of gate.

The AND Gate

An AND gate, shown below, has at least two input terminals. The output of an AND gate will be **1** (*i.e.*, equal to 5 V relative to ground) if *all* of its inputs are **1**. If one or more inputs is **0**, then the AND gate's output will be **0**. The circuit symbol and truth table for a two-terminal AND gate are shown below. The table contains four rows because its two inputs have precisely four (2^2) possible combinations of values.

[1] In some logic systems, particularly those found in battery-operated devices, logic voltages levels of 0 V and 3 V are used to save power

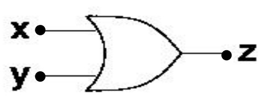


Inputs			Output
x	y		z
0	0		0
0	1		0
1	0		0
1	1		1

Two-Input AND Gate   AND-Gate Truth Table

The OR Gate

The OR gate has two or more inputs and a single output whose value is **1** if *any* its inputs is **1**. Otherwise, the gate output will be **0**. Said another way, the output of an OR gate will be **0** if *all* its inputs are **0**.



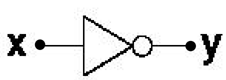
Two-Input AND Gate
(common ground not shown)

Inputs			Output
x	y		z
0	0		0
0	1		1
1	0		1
1	1		1

OR-Gate Truth Table

The NOT Gate

The NOT gate, or *logical inverter*, has only one input and one output. The value of the output is equal to the logical opposite of its input. Thus the output of a NOT gate is **1** if the input is **0**. Conversely, the output will be **0** if the input is **1**. :



x		y
0		1
1		0

The NOT Gate

The NAND Gate

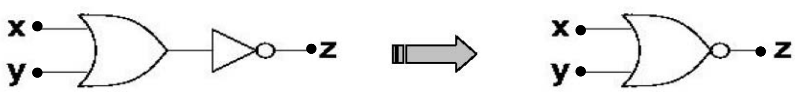
Sometimes it is useful and more practical to combine functions of the AND and OR gates with the NOT gate to produce slightly more complex gates. The NAND consists of an AND gate in series with a NOT gate. The abbreviated symbol for the above consists of the AND gate symbol with a hollow circle attached to its "nose". The truth table for a two-input NAND gate is identical to that of a two-input AND gate except that its output values are inverted.

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

Two-Input NAND and Truth Table

The NOR Gate

The NOR gate similarly consists of an OR gate in series with a NOT gate: The truth table for a two-input NOR gate is identical to that of a two-input OR gate except that its output values are inverted:

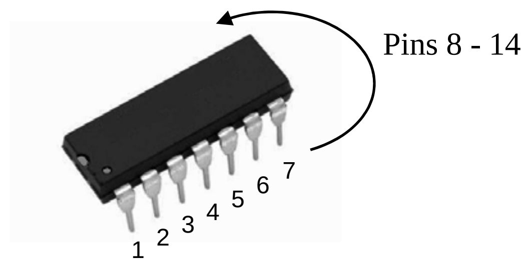


Two-Input NOR and Truth Table

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

Finding Data Sheets for Logic Gates

Your EK307 parts kit contains a variety of logic gates in the form of integrated-circuit "DIP" chips. The standard DIP, or *dual-inline package,* can have anywhere from 6 to 20 pins. A typical 14-pin variety is shown to the right. The indents at the left end serve as pin locators; your chip may have one or both of these identifiers. Pins are numbered counter-clockwise around the chip beginning at pin 1.



Datasheets for most common logic gates in DIP packages can be found on the Internet. One of your tasks in this lab is to find the datasheet for the parts you wish to use and apply the information to your design.

Using the Breadboard

The *solderless breadboard* is a tool for the rapid prototyping of circuits. The breadboard contained in your kit is typical of most versions. Skill in its use is essential for the practicing engineer. As you can see in Fig. 1, the surface of board is covered by small holes, each of which accommodates the insertion of a single wire or component lead. Beneath the plastic cover, the holes are connected into various groups by a network of metal pathways. Specifically, each of the five holes in the short rows are interconnected, as are each of the long rows on the edges of the board. This interconnection pattern is shown in Fig. 2. The short rows serve as the nodes of a circuit built on the breadboard; the long, outer rows are usually used to distribute power, ground, and sometimes signals to the various sections of the circuit. The layouts of some typical circuits that include integrated circuit "chips" are shown in Figs. 3 -5.

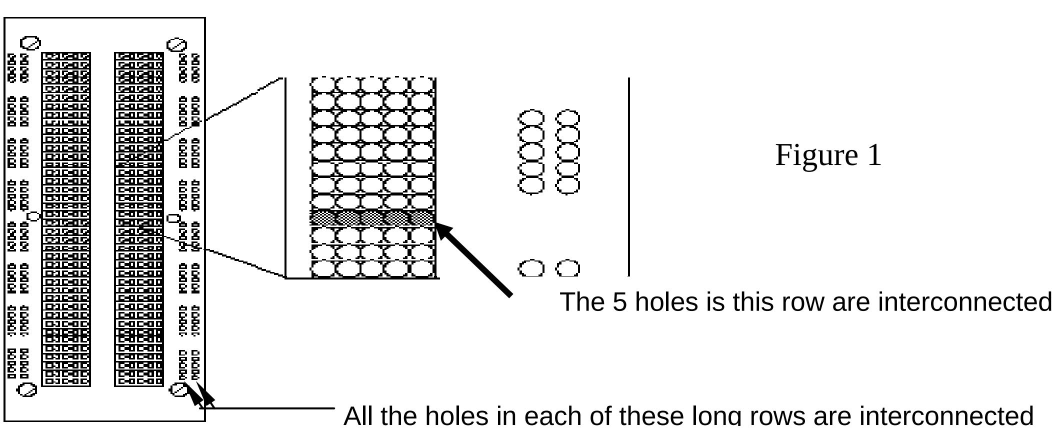


Figure 1

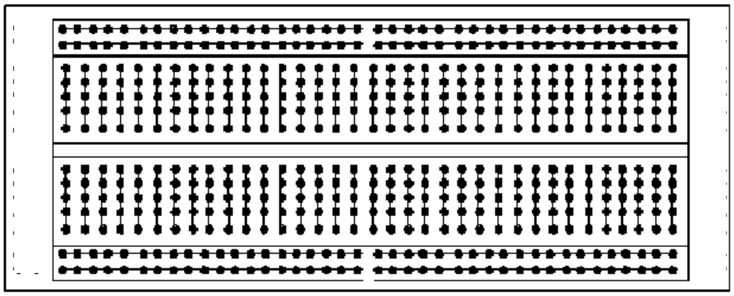


Figure 2

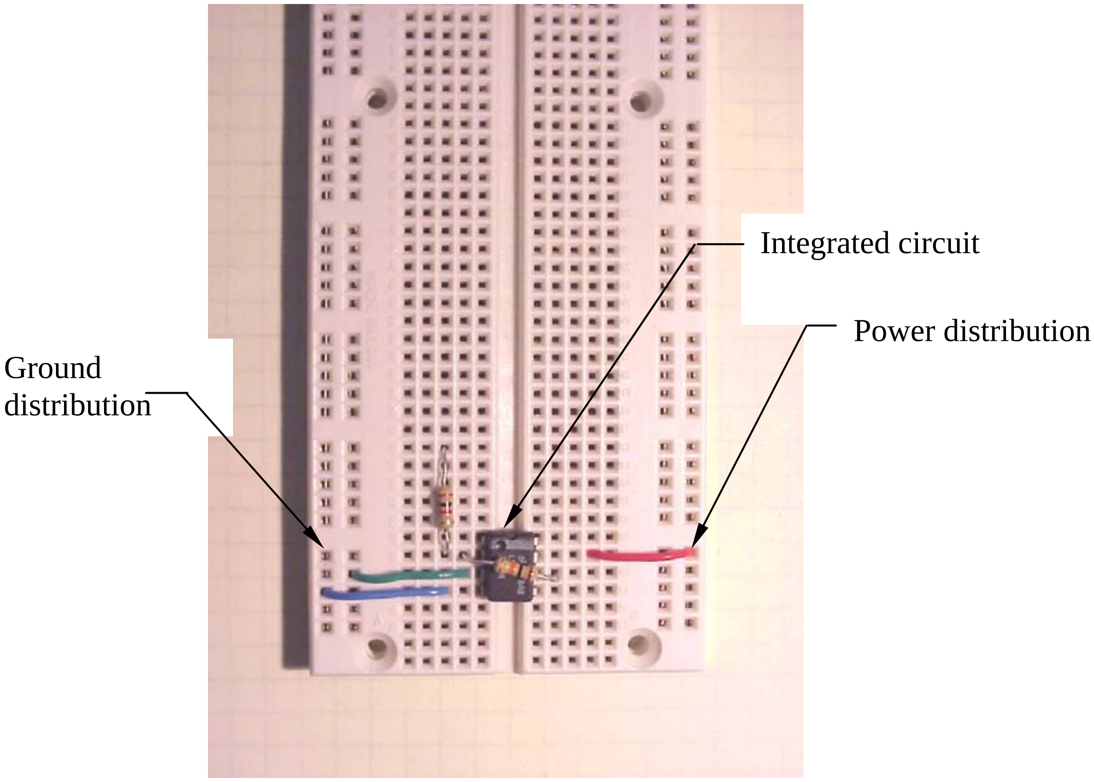


Figure 3

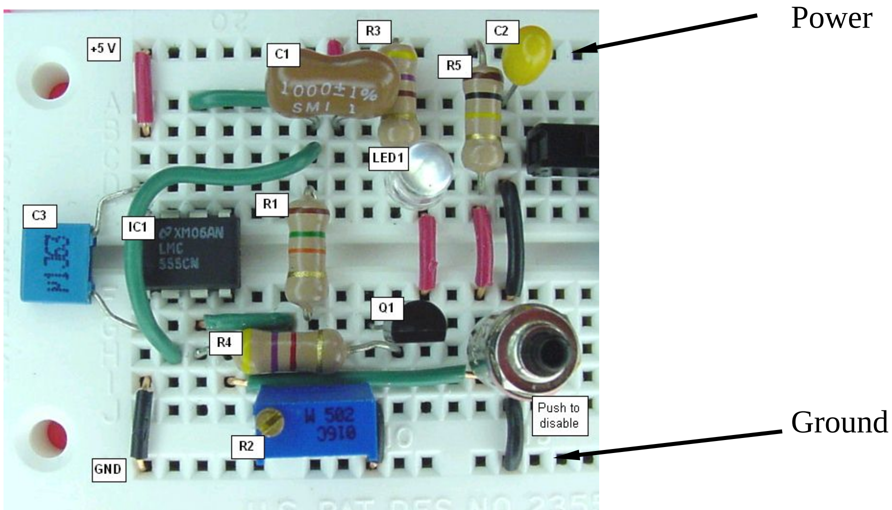


Figure 4

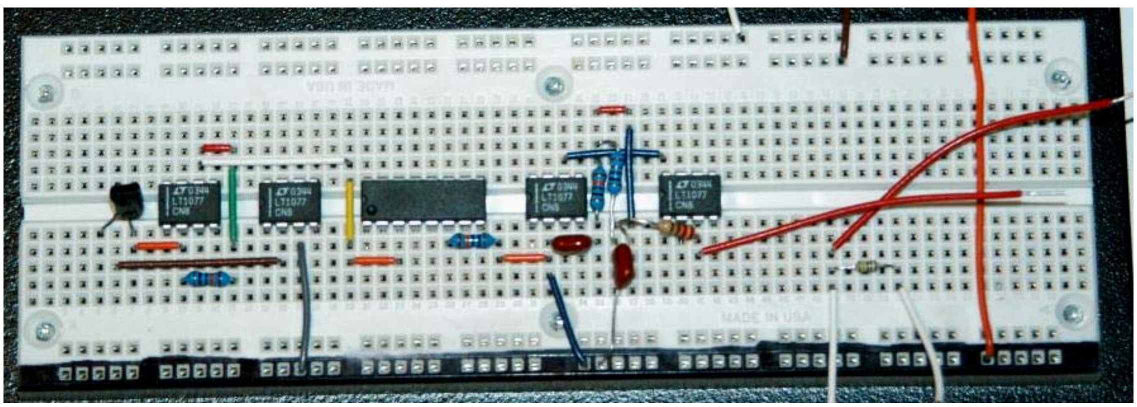


Figure 5

Avoiding the "Bird's Nest" Pitfall.

Good wiring practice requires that a breadboard circuit be compact, neat, and orderly, with all leads cut as short as possible. As shown in the above figures, component bodies should physically rest on or just above the board surface, and wires should be easy to trace and touch with a probe. The "bird's nest" approach of Figs. 6 and 7, in which wires dangle and go haphazardly in every direction, should be avoided at all costs. Such a disorderly tangle of wires can cause component leads to short together causing wiring errors. Circuit testing also becomes extremely difficult when a circuit is messy as one become easily lost in a chaotic circuit. A sloppy circuit affects the attitude of the engineer, who is likely to take the design or analysis task less seriously if work on the circuit is difficult. The wise engineer produces circuits that are neat, compact, tidy, and easily accessible.

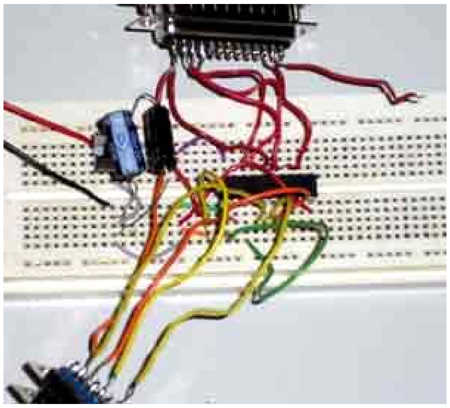


Figure 6

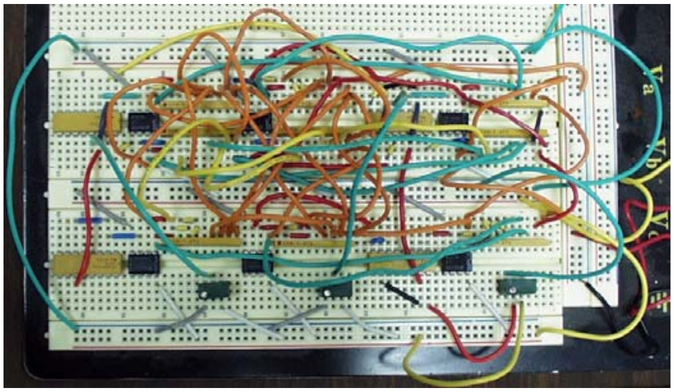


Figure 7

Lab Assignment

Level 1

Build a logic circuit that indicates when a unanimous vote occurs for five inputs. Simulate votes by a wire or switch that connects a given gate input to 5-V (logic **1**, or "yes") or to ground (logic **0**, or "no"). Use one or more LEDs to indicate the status of the vote result.

Level 2

Build a logic circuit that indicates when a majority "yes" vote occurs for three voters. Simulate votes by a wire or switch that connects a given gate input to 5-V (logic **1**, or "yes") or to ground (logic **0**, or "no"). Use one or more LEDs to indicate the status of the vote result.

Level 3

Build a logic circuit that converts a 3-bit, base-2 binary input (e.g., the binary numbers **000** through **111**) to one of eight LED outputs. This function is summarized by the following diagram: