ELG3331: Lab 3
Digital Logic Circuits

What does Digital Means?

Digital describes any system based on discontinuous data or events. Typically digital is computer data or electronic sampling of an analog signal. Computers are digital machines because at their most basic level they may distinguish between just two values, 0 and 1, or off and on. All data that a computer processes must be encoded digitally, as a series of zeroes and ones.

The opposite of digital is analog. A typical analog device is a clock in which the hands move continuously around the face. Such a clock is capable of indicating every possible time of day. In contrast, a digital clock is capable of representing only a finite number of times (every tenth of a second, for example). In general, humans experience the things analogically. Vision, for example, is an analog experience because we perceive infinitely smooth gradations of shapes and colors. Most analog events, however, can be simulated digitally. Photographs in newspapers, for instance, consist of an array of dots that are either black or white. From afar, the viewer does not see the dots (the digital form), but only lines and shading, which appear to be continuous. Although digital representations are approximations of analog events, they are useful because they are relatively easy to store and manipulate electronically. The trick is in converting from analog to digital and back again.

Traditionally, digital means the use of numbers and the term comes from digit, or finger. Today, digital is synonymous with computer.

Logic Gates

A gate is an electronic circuit that performs a logical operation. Logic gates are the basic building blocks for digital electronic circuits. A switching circuit or circuit is a composition of gates. Operations on the binaries 0, 1 may be viewed as truth functional operations, and binary arithmetic as an application of propositional logic. Binary arithmetic is implemented by building switching circuits.

A switch may represent a two-state logic condition. An open switch will be indicated by a 0, while a closed switch will be indicated by a 1. Figure 1 shows a simple switch

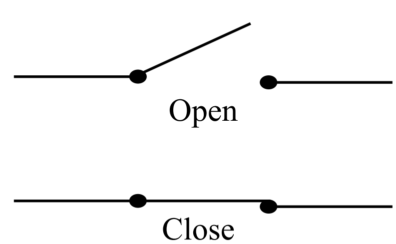


Figure 1 Two conditions for a simple switch.

Logic Expression

Let us consider the circuit of Figure 2. The output condition (f_o) of the light (on or off) depends on the condition of the switch. For example, light = 1, dark = 0. Table 1 defines the operation of the circuit. This type of table is called a **truth able**. The lamp will produce light when the switch is closed (f_o = 1).

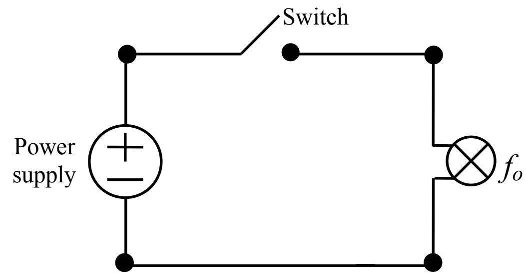


Figure 2 Electric circuit with a switch.

Switch condition	f_o
0	0
1	1

Table 1 Truth Table.

Logic Operators

There are three basic logic operations which are used in the design of all logic circuits. These logic operations are:

1. The OR operator which is indicated by a plus sign (+) as: A + B
2. The AND operator which is indicated by a dot or multiplication sign (×): A . B or A × B.
3. The NOT operator which is indicated by an overbar as: $\overline{A}$.

The AND Operator/Gate

The AND operator implements the AND function. It depends upon two or more events happening at the same time. The digital device used for this operation is called the AND gate. With all inputs must have logic 1 signals in order for the output to be logic 1. With either input at logic 0, the output will be held to logic 0. We may visualize the AND gate as an electrical circuit involving two switches in series as shown in Figure 2.

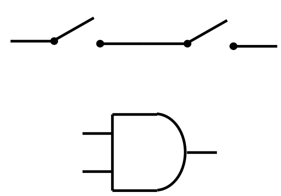


Figure 2 AND gate representation.

An example of an AND gate is an interlock control system for a machine tool such that if the safety guard is in place and gives a logic 1 and the power is on, giving a logic 1, then there can be an output of 1. Accordingly, the machine will operate.

With the two-input AND gate shown in Table 15-6, there are 2^2 or 4 possible combinations of the variables each variable has two possible values.

Logical AND: If both inputs X and Y are 1, then the output Z is 1

There is no limit to the number of inputs that may be applied to an AND function, so there is no functional limit to the number of inputs an AND gate may have. However, for practical reasons, commercial AND gates are most commonly manufactured with 2, 3, or 4 inputs. A standard IC package contains 14 or 16 pins, for practical size and handling. A standard 14-pin package can contain four 2-input gates, three 3-input gates, or two 4-input gates, and still have room for two pins for power supply connections.

The OR Operator/Gate

The OR operator is sort of the reverse of the AND operator. In symbols, the OR function is designated with a plus sign (+). The OR function, like its verbal counterpart, allows the output to be true (logic 1) if any one or more of its inputs are true. The digital device used for this operation is called the OR gate. Like the AND gate, the OR gate has 2^n possible combinations required to describe its operation (n is the number of inputs). We may visualize the OR gate as an electrical circuit involving two switches in series as shown in Figure 3.

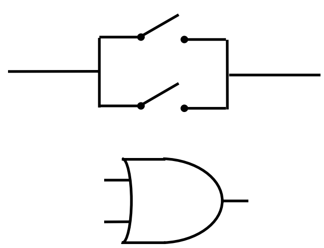


Figure 3 OR gate representation.

Logical OR: If either inputs X or Y are 1, then the output Z is 1

As with the AND function, the OR function can have any number of inputs. However, practical commercial OR gates are mostly limited to 2, 3, and 4 inputs, as with AND gates. With the two-input OR gate shown in Table 15-6, there are 2^2 or 4 possible combinations of the variables each variable has two possible values.

The NOT Operator or Inverter

The function of the NOT operator (gate) may be understood by studying the behavior of the switch shown in Figure 4. The switch is a two-state device, however, it can be in only one state at a time. For example, the logic number 1 represents the switch in the closed position, while the logic number represents the open position of the switch.

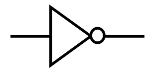


Figure 4 NOT gate representation.

The NOT gate which is called also an **inverter** is the simplest logic gate. It is a little different from AND and OR gates in that it always has exactly one input as well as one output. Whatever logical state is applied to the input, the opposite state will appear at the output. The NOT function is denoted by a horizontal bar over the value to be inverted, as shown in Table 15-6. In the inverter symbol, the triangle actually denotes only an amplifier, which in digital terms means that it "cleans up" the signal but does not change its logical sense. It is the circle at the output, which denotes the logical inversion. The circle could have been placed at the input instead, and the logical meaning would still be the same.

The NAND Gate

The NAND gate is the complementary form of the AND gate. It is very commonly used in practice. It is verified that the logic function implemented by NAND gate corresponds to AND gate followed by an inverter. Another way of considering the NAND gate is as an AND gate with a NOT gate applied to invert both the inputs before they reach the AND gate. When input A is 1 and input B is 1 there is an output of 0, all other inputs giving an output of 1.

The NOR Gate

The NOR gate is the complementary form of the OR gate. It is verified that the logic function implemented by NOR gate corresponds to OR gate followed by an inverter. Another way of considering the NOR gate is as an OR gate with a NOT gate applied to invert both the inputs before they reach the OR gate. When input A or input B is 1 there is an output of 0.

Half Adder

The half adder (Figure 5) is an example of a simple, functional digital circuit built from two logic gates. The half adder adds two one-bit binary numbers (AB). The output is the sum of the two bits (S) and the carry (C). The same two inputs are directed to two different gates. The inputs to the XOR gate are also the inputs to the AND gate.

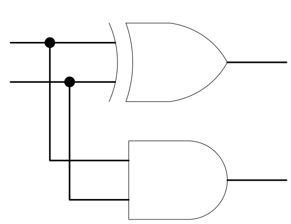


Figure 5 Half Adder

Full Adder

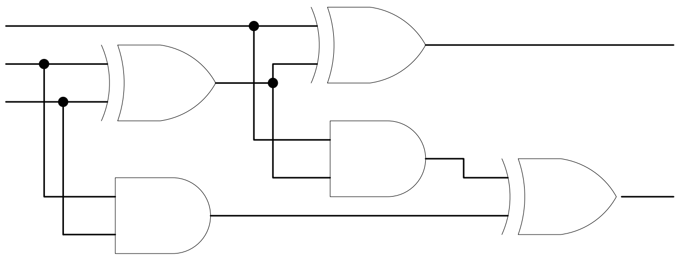
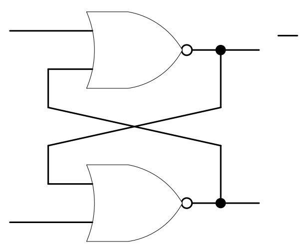


Figure 6 Full Adder

The full-adder circuit (Figure 6) adds three one-bit binary numbers (C A B) and outputs two one-bit binary numbers, a sum (S) and a carry (C_1). If you look closely, you'll see the full adder is simply two half adders joined by a XOR.

SR Flip Flop



C

A

B

Figure 7 SR Flip Flop

An SR Flip Flop (Figure 7) is an arrangement of logic gates that maintains a stable output even after the inputs are turned off. This simple flip flop circuit has a set input (S) and a reset input (R). The set input causes the output $Q = 1$ and $\overline{Q} = 0$. The reset input causes the opposite to happen, $Q = 0$ and $\overline{Q} = 1$. Once the outputs are established, the state of the circuit is maintained until S or R goes high, or power is turned of to the circuit.
This is a simple model of how a bit of RAM can be perpetuated.

Experiment with Logic Gates

Initial Procedure

- Use the breadboard, the IC's and the jumper wires.
- Connect the +5 V (V_{CC}) and ground connections to the IC.
- Connect the output signals to the BNC connectors on the breadboard for the subsequent feeding to the Oscilloscope or the DMM's.
- The IC input signals can be connected to the fixed supply voltage, where appropriate, or, where HI or LO transitions are desired.

Exercise

1. Verify the truth table of the following two circuits:

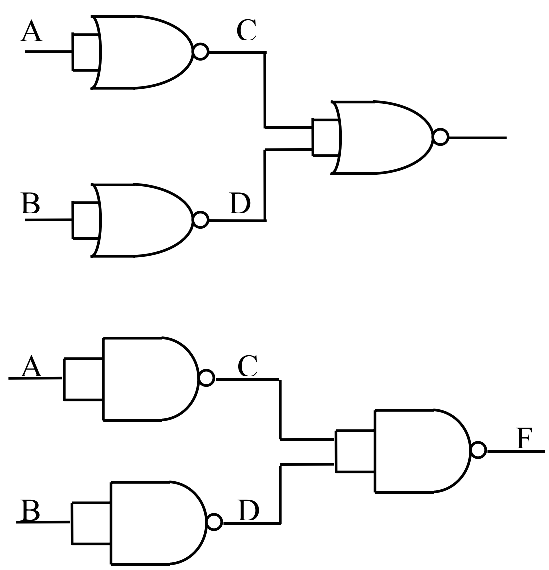


Figure 8 Three NOR and three NAND circuits.

2. Verify the truth table of the Half Adder.

3. Verify the truth table of the SR Flip Flop.

Design 1

- Design or assemble an OR gate using NAND gates only.
- A NAND gate can provide INVERTER operation by connecting both inputs together.
- To facilitate the design, refer to de Morgan's theorem (page 627 of the textbook): $\overline{A+B} = \overline{A}.\overline{B}$
- Draw a diagram for the circuit and verify your results by constructing the truth tables for the circuit obtained.

Design 2

Show a way to make a three-input OR gate out of two-input OR gates.