PACE Electrical Engineering

Digital Logic

Introduction

In a previous lesson we learned a little about Boolean Algebra, a branch of mathematics that formalizes logical thinking. We learned about logical operations, such as AND, OR, and NOT. These functions are characterized by truth tables, which determine the truth of a statement based on the truth of statements on which it depends. So, for example, if A, B, and C are the following statements:

A: I have good attendance at PACE

B: I pay attention in EE class

C: I do very well on my EE projects

then C is dependent on A and B by the function C = AB. (The statement 'I do very well on my EE projects' is true, if 'I have good attendance at PACE' is true AND 'I pay attention in EE class' is true.). The truth table for C determines its truth value for every combination of truth values for A and B.

Principles of Boolean Algebra can be implemented in electronics to produce circuits that can make logical decisions, do calculations, and perform complex operations. Circuits based on these principles are called digital circuits and use digital logic.

Digital Logic Basics

Digital Logic is an abstraction that can be implemented with electronic circuits using resistors, transistors and other elements that we have learned about. The basis of this abstraction is a set of structures called logic gates. Logic gates perform Boolean operations such as AND, OR, NOT, and others. Each of these gates has one or more inputs and one output. Their input-output relationships are characterized by truth tables.

Terminology

In Boolean Algebra we discussed statements which were either true or false. We used logical variables to represent these statement and the logical operators dealt with these variables. Although Digital Logic is essentially the same as Boolean Algebra, the terminology differs. Logic gates deal with **signals,** which have values of either 0 or 1. These values are equivalent to false (for 0) and true (for 1). However, they can represent anything that can be in one of two states: on or off, high or low, open or closed. Which it represents depends on the type of circuit under consideration. Logic gates are also characterized by truth tables, but use 1 and 0 in place of T and F. For example, we will now show the truth table for a two-input AND function as

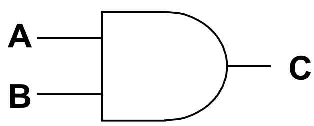
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table for AND Function

Logic Gates

The building blocks for digital circuits are **logic gates**. This section describes several logic gates. It gives the name, the symbol, the truth table, and a description.

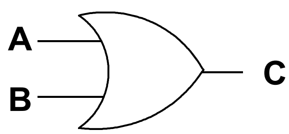
AND Gate



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

The output is 1 only if both inputs are 1. It's 0 otherwise.

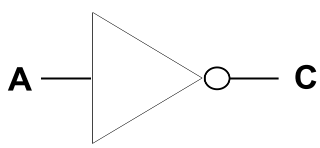
OR Gate



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

The output is 1 if either or both inputs are 1. It's 0 only if both inputs are 0.

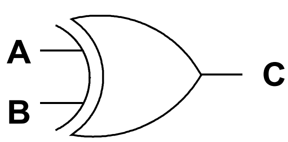
Inverter



A	C
0	1
1	0

The output is 1 if the input is 0. The output is 0 if the input is 1.

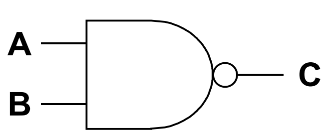
XOR Gate (Exclusive OR)



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

The output is 1 only if exactly one of the inputs is 1. It's 0 only i.f both inputs are 0 or both are 1.

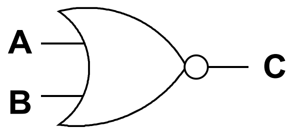
NAND Gate



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

The output is 0 only if both inputs are 1. It's 1 otherwise. The NAND is equivalent to an AND gate followed by an Inverter. (Same as NOT AND)

NOR Gate



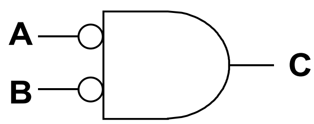
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

The output is 1 only if both inputs are 0. It's 0 otherwise. The NOR is equivalent to an OR gate followed by an Inverter. (Same as NOT OR)

Conventions

Circles Represent Inverters

As you may have figured out from the examples above, a circle in a gate indicates inversion (or negation). They are a shorthand way in representing an inverter. In addition to the circle at the output as we saw above, circles can be at the input indicating that the inputs are inverted. For example, examine this AND gate with inverted inputs:



Mathematical Operations Denote Logical Operations

Mathematical operations multiplication and addition are used to represent logical operations. AND is indicated by multiplication and OR is indicated by addition. If A and B are the inputs to an AND gate and C is the output. The AND operation is represented by

$$C = AB$$

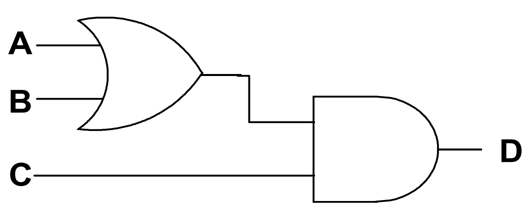
The OR operations is

$$C = A + B$$

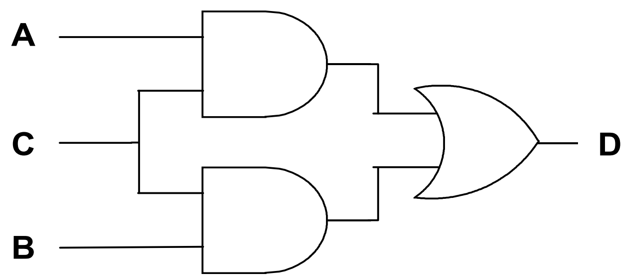
Inversion is represented by a bar over the input variable. For the inverter:

$$C = \overline{A}$$

Addition and multiplication are used because the corresponding logical operations combine in the same way as these mathematical operations. For example, the following circuit



is represented by $D = (A + B)C$. Using normal math, this equation is equivalent to $D = AC + BC$, which is implemented by the following circuit:



If you create the truth tables for each circuit you can prove that they are equivalent.