Boolean Algebra

Digital computers contain circuits that implement Boolean functions.

Boolean algebra is a mathematical system for the manipulation of variables that can have one of two values ("true" and "false")

Boolean expressions are created by performing operations on Boolean variables.

Common Boolean operators include AND, OR, and NOT.

Boolean Algebra

In digital systems, these values are “on” and “off,” 1 and 0, or “high” and “low.”

This is why the binary numbering system is a natural basis for digital systems.

Logic circuits take one or more Boolean or digital inputs and generate a digital output.

We will examine how logic circuits can be made up from combinations of basic logic gates.

Boolean Algebra

A Boolean function has:

- At least one Boolean variable,
- At least one Boolean operator, and
- At least one input from the set {0,1}.

It produces an output that is also a member of the set {0,1}.

Boolean functions can be represented using truth tables and can be implemented using logic gates

The truth table for the Boolean function:

$$F(x,y,z) = xz' + y$$

is shown at the right.

The extra (shaded) columns hold evaluations of subparts of the function.

$$F(x,y,z) = xz' + y$$

x	y	z	z'	xz'	xz' + y
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	1

Like arithmetic operators, Boolean operators have rules of precedence.

The NOT operator has highest priority, followed by AND and then OR

z' denotes NOT z

xz denotes x AND y

x + y denotes x OR y

$F(x,y,z) = xz' + y$

x	y	z	z'	xz'	xz' + y
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	1

- The simpler the Boolean functions, the fewer the number of logic gates needed to implement them.
- Simpler circuits are cheaper to build, consume less power, and run faster than complex circuits.
  - Best to reduce our Boolean functions to their simplest form.
- There are a number of Boolean identities that help us to do this.

Most Boolean identities have an AND (product) form as well as an OR (sum) form. We give our identities using both forms. The group below is rather intuitive:

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0 + x = x$
Null Law	$0x = 0$	$1 + x = 1$
Idempotent Law	$xx = x$	$x + x = x$
Inverse Law	$xx' = 0$	$x + x' = 1$

This second group of Boolean identities should be familiar to you from your study of algebra:

Identity Name	AND Form	OR Form
Commutative Law	xy = yx	x+y = y+x
Associative Law	(xy)z = x(yz)	(x+y)+z = x + (y+z)
Distributive Law	x+yz = (x+y)(x+z)	x(y+z) = xy+xz

The next group of Boolean identities are perhaps the most useful.

If you have studied set theory or formal logic, these laws are also familiar to you.

Identity Name	AND Form	OR Form
Absorption Law	$x(x+y) = x$	$x + xy = x$
DeMorgan's Law	$(xy)' = x' + y'$	$(x+y)' = x'y'$
Double Complement Law	$(x)'' = x$	