Introduction to Boolean logic and Logical Gates

Aniel Nieves-González

Institute of Statistics

Fall 2014

- We saw the importance of the binary number system for data representation in a computer system.
- We'll see that the construction of computer circuits is based upon a branch of mathematics called Boolean algebra.

Boolean algebra

- Boolean algebra consists of the set $\{0, 1\}$ with the operations complement ($'$), sum ($+$), and product ($\cdot$).
- We can rephrase the previous definition as: the set $\{\text{FALSE}, \text{TRUE}\}$ with the operations of negation (NOT, $\sim$), disjunction (OR, $\wedge$), and conjunction (AND, $\vee$).
- A Boolean variable is one that assumes one of the two aforementioned possible values.

Boolean algebra

Let x, y, and z be Boolean variables:

- The *NOT* or *complement* operation is defined as:

x	x'
1	0
0	1

- The *AND* or *product* is defined as:

x	y	$x \cdot y$
1	1	1
1	0	0
0	1	0
0	0	0

- The *OR* or *sum* operation is defined as:

x	y	$x+y$
1	1	1
1	0	1
0	1	1
0	0	0

- Boolean expressions take values on $\{0,1\}$. E.g. $x+y$ or $z(x+y)$.
- As with the definition of the basic operations, Boolean expressions can be represented with a *truth table*.
- If a Boolean expression has n variables the table will have 2^n rows and $n+1$ columns. The first n columns define the posible values of the variables and the $n+1$ column the result for the expression.

Properties of Boolean algebra. Let x, y, and z be Boolean expression.

- Identity of the sum, $x + 0 = x$.
- Identity of the product, $x \cdot 1 = x$.
- Domination law of the sum, $x + 1 = 1$.
- Domination law of the product, $x \cdot 0 = 0$.
- Conmutativity of the sum, $x + y = y + x$.
- Conmutativity of the product, $x \cdot y = y \cdot x$.
- Asociativity of the sum, $x + (y + z) = (x + y) + z$.
- Asociativity of the product, $x \cdot (y \cdot z) = (x \cdot y) \cdot z$.

- Distributivity of the product with respect of the sum, $x \cdot (y + z) = x \cdot y + x \cdot z$.
- Distributivity of the sum with respect of the product, $x + (y \cdot z) = (x + y) \cdot (x + z)$.
- De Morgan law of the sum, $(x + y)' = x' \cdot y'$.
- De Morgan law of the product, $(x \cdot y)' = x' + y'$.

Boolean algebra

Even more properties:

$$x + x' = 1$$

$$x \cdot x' = 0$$

$$(x')' = x$$

$$x + x = x$$

$$x \cdot x = x$$

$$x + (x' \cdot y) = x + y$$

$$x \cdot (x' + y) = x \cdot y$$

Boolean algebra

Simplifying Boolean expressions:

$$\begin{aligned}
x + (x' \cdot y) &= (x + x') \cdot (x + y) && \text{distributive property} \\
& && \text{of the sum over the product} \\
&= 1 \cdot (x + y) && \text{complement of sum} \\
&= x + y && \text{identity of product}
\end{aligned}$$

$$\begin{aligned}
x \cdot (x' + y) &= (x \cdot x') + (x \cdot y) && \text{distributive property} \\
& && \text{of product over sum} \\
&= 0 + (x \cdot y) && \text{complement of the product} \\
&= x \cdot y && \text{identity of the sum}
\end{aligned}$$

- Simplify the following expressions, verify with the truth table.

$$x(x' + z) + xz'$$

  (Answer: x.)

- Verify the following identities:

$$y' + xy + x'y = 1$$

$$z'(y + (x + y)(x + z) + xy) = xz' + yz'$$

- We know how to construct the truth table for a Boolean expression. But, how to go the other way around.
- Let F be a Boolean function of n variables $x_1, \dots, x_n$, a boolean product in which each one of the variables appears once is called miniterm.
- For a function of three variables (x, y and z) the possible miniterms are:

$$xyz, xyz', xy'z, x'yz, xy'z', x'yz', x'y'z, x'y'z'.$$

Boolean algebra

- Observe that a boolean function of n variables could have up to 2^n different miniterms.
- Each term represents a row in the truth table.
- If the value of the variable is 0 then the complement of the variable appears in the term.
- If the value of the variable is 1 then the variable itself appears in the term.

Boolean algebra

- Any Boolean function can be written as a sum of miniterms. This way of writing the Boolean function is called the sum of product.
- To find the sum of products form of a Boolean function from the truth table choose the sum of miniterms for which $F = 1$.

For example, consider the following truth table for the Boolean function F.

x	y	F	miniterm
0	0	1	$x'y'$
0	1	0	$x'y$
1	0	1	xy'
1	1	1	xy

The sum of products is given by the sum of miniterms that correspond to the rows with value 1.

$$F(x, y) = x'y' + xy' + xy.$$

Let $F(x, y, z) = (x + y')z$. Determine the sum of products of F.
(Answer: $F(x, y, z) = xyz + xy'z + x'y'z$.)

We can reach the answer using the properties that we already know:

$$
\begin{aligned}
F(x,y,z) = & \quad (x+y')z & = xz + y'z \\
= & \quad (1)xz + (1)y'z & = (y+y')xz + (x+x')y'z \\
= & \quad yxz + y'xz + xy'z + x'y'z & = yxz + xy'z + x'y'z
\end{aligned}
$$

$$\therefore F(x,y,z) = (x+y')z = xyz + xy'z + x'y'z$$

We can use a truth table approach:

x	y	z	y'	$x+y'$	F	miniterm
0	0	0	1	1	0	$x'y'z'$
0	0	1	1	1	1	$x'y'z$
0	1	0	0	0	0	$x'yz'$
0	1	1	0	0	0	$x'yz$
1	0	0	1	1	0	$xy'z'$
1	0	1	1	1	1	$xy'z$
1	1	0	0	1	0	xyz'
1	1	1	0	1	1	xyz

$$\therefore F(x,y,z) = xyz + xy'z + x'y'z$$

- The complement of a miniterm is its respective maxiterm.
- Such complement is obtained using De Morgan laws and it is called the product of sums.
- To find the product of sums form of a Boolean function from a truth table, we choose the sum of miniterms for which $F = 0$ and take the complement of the result applying the De Morgan laws.

Boolean algebra

Let us return to the truth table of the previous example.

x	y	F	miniterm
0	0	1	$x'y'$
0	1	0	$x'y$
1	0	1	xy'
1	1	1	xy

We take the complement of the sum of miniterms that correspond to the rows with value 0.

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

applying the De Morgan laws.

Exercise. Consider the following truth table and construct the boolean function:

x	y	F	miniterm
0	0	0	
0	1	1	
1	0	1	
1	1	0	

Exercise. Consider the following truth table and construct the boolean function:

x	y	F	miniterm
0	0	0	$x'y'$
0	1	1	xy'
1	0	1	$x'y$
1	1	0	xy

$$\therefore F(x, y) = xy' + x'y$$

This function is called the "exclusive or" (XOR).

Two important boolean operations:

- NOR means NOT OR, i.e., sum's complement

$$x \downarrow y = (x+y)' = x'y'$$

- NAND means NOT AND, i.e., product's complement.

$$x \uparrow y = (xy)' = x' + y'$$

The truth tables for those operations are:

x	y	$x \downarrow y$	$x \uparrow y$
1	1	0	0
1	0	0	1
0	1	0	1
0	0	1	1

All boolean functions can be represented with the operators NOR or NAND.

- For example: the complement, sum, and product can be expressed with the NOR operator.

$$x' = x \downarrow x$$

$$xy = (x \downarrow x) \downarrow (y \downarrow y)$$

$$x + y = (x \downarrow y) \downarrow (x \downarrow y)$$

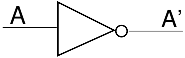
Logical gates and logical circuits

- A *gate* is an electronic device that operates on a collection of binary inputs and produces a binary output. That is, it maps from the set $\{0, 1\}$ (or a set of n-tuples $\{0, 1\} \times \ldots \times \{0, 1\}$) to $\{0, 1\}$.
- An unary operator has one input and one output. A binary operator has two inputs and one output.
- We'll discuss the logical gates for the basic operations: OR, AND, NOT.

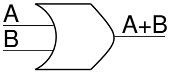
Logical gates and logical circuits

The logical gates are represented graphically as follows:

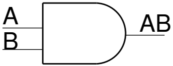
- The complement (NOT)



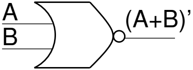
- The sum (OR)



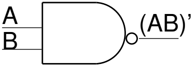
- The product (AND)



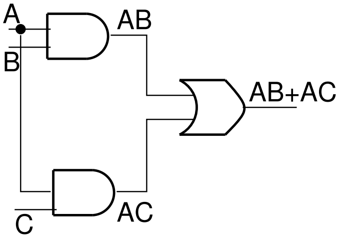
- NOR



- NAND



A logical circuit is a combination of two or more logical gates. The gates are combined by using the output of a gate as the input of another. For example:



Logical circuits: Example

Design a logica circuit for the function:

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

A light bulb is controlled by two electrical switches. The light bulb will turn on when both switches are on. Any other combination will turn off the light bulb. If the light bulb is on, it can be turned off by turning off any switch. Design a circuit that implements how the light bulb functions.

- Let x and y be the variables that represent the switches and let $F(x, y)$ represent the state of the light bulb.
- Assign 1 to the state when the switch is on, and 0 to state when the state is off.
- Observe that the light bulb is on when both switches are on.

Logical circuits: Example

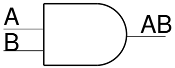
- We construct a truth table to obtain the function.

x	y	$F(x,y)$
1	1	1
1	0	0
0	1	0
0	0	0

- By writing the sum of products we obtain the function

$$F(x,y) = xy$$

- The circuits that implements the light bulb's function is:



Logical circuits: Example

In amateur boxing to assign points to the boxers the following method is used: Three judges assign points. Each judge press a button when he/she see that one boxer lands a punch in the other. To assign a point to the boxer at leasr two of the three judges have to press the button simultaneously. That is, a boxer receives a point when the majority of the judges agree that the boxer hits the adversary. Design a circuit to solve this problem.

Logical circuits: Example

- Let x, y, and z be the values that assign each judge in a given moment and $F(x, y, z)$ be the result whether a point is given to a boxer.
- The truth table is

x	y	z	$F(x, y, z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Circuitos lógicos: Ejemplo

- Writing the sum of products functions

$$\begin{aligned} F(x,y,z) &= x'yz + xy'z + xyz' + xyz \\ &= (x'y + xy' + xy)z + xyz' \\ &= (x'y + x(y' + y))z + xyz' \\ &= (x'y + x)z + xyz' \end{aligned}$$