Cheatsheet - Logic Gates

Fabio Lama – fabio.lama@pm.me

1. Intro

(NOTE: Reading the *Postulates of Boolean Algebra* cheatsheet is recommended here)

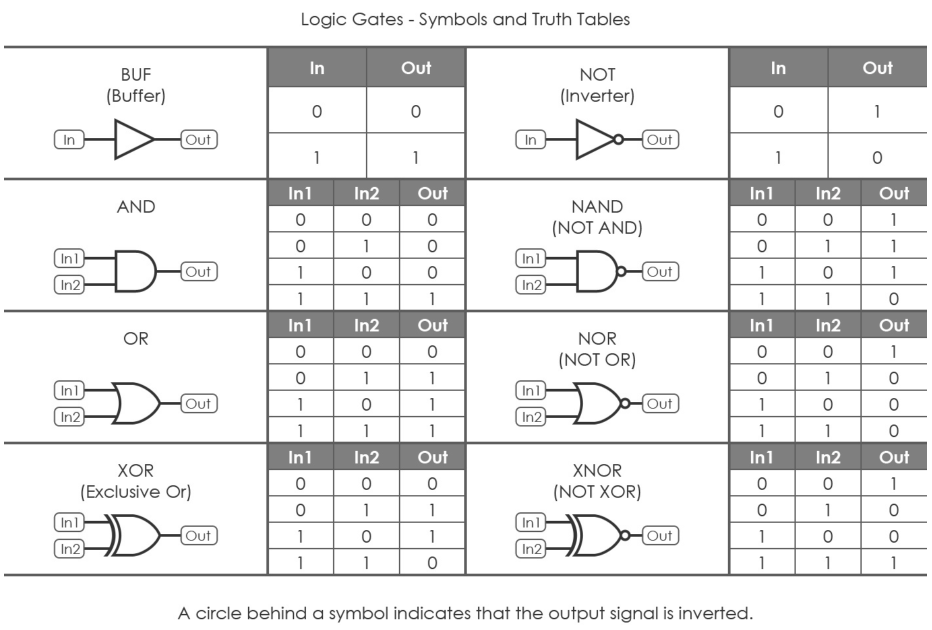
Logic gates are basic elements of circuits implementing Boolean operations. The most basic circuits are **OR** gates, **AND** gates and invertors (**NOT** gates). All boolean functions can be written in terms of these three logic operations.

- **AND** operation is represented as $f = x.y$ or $f = xy$.
- **OR** operation is represented as $f = x + y$.
- **NOT** operation is represented as $f = \bar{x}$.

Other gates:

- **XOR** operation is *true* only when the value of the inputs differ.
- **NAND** operations is equivalent to "not AND".
- **NOR** operation is equivalent to "not OR".
- **XNOR** operation is equivalent to a "not XOR".

AND, OR, XOR and XNOR are **commutative** (e.g. $a + b = b + a$) and **associative** (e.g. $a + (b + c) = (a + b) + c$). NAND and NOR are commutative but not associative.



Logic Gates - Symbols and Truth Tables

BUF (Buffer)

In	Out
0	0
1	1

NOT (Inverter)

In	Out
0	1
1	0

AND

In1	In2	Out
0	0	0
0	1	0
1	0	0
1	1	1

NAND (NOT AND)

In1	In2	Out
0	0	1
0	1	1
1	0	1
1	1	0

OR

In1	In2	Out
0	0	0
0	1	1
1	0	1
1	1	1

NOR (NOT OR)

In1	In2	Out
0	0	1
0	1	0
1	0	0
1	1	0

XOR (Exclusive Or)

In1	In2	Out
0	0	0
0	1	1
1	0	1
1	1	0

XNOR (NOT XOR)

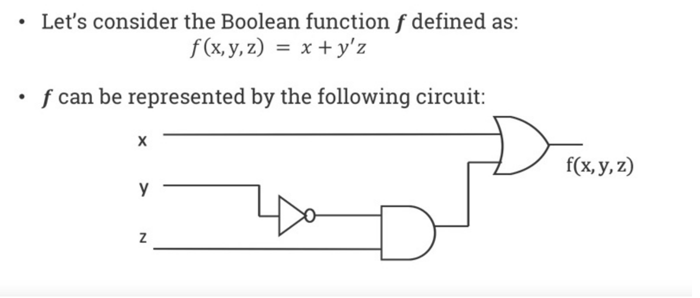
In1	In2	Out
0	0	1
0	1	0
1	0	0
1	1	1

A circle behind a symbol indicates that the output signal is inverted.

Figure 1. Source: http://www.exclusivearchitecture.com/?page_id=2425

2. Circuits

We describe the combination of logic gates as a **circuit**.



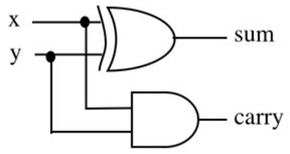
- Let's consider the Boolean function ***f*** defined as:
  $f(x, y, z) = x + y'z$
- ***f*** can be represented by the following circuit:

A circuit that's used for the **addition** of inputs is called an **adder**. A **half adder** takes two inputs and generates a **carry** and a **sum**. A **full adder** takes three inputs and generates a carry and a sum.

For example, an **half adder**:

$$\text{sum} = xy' + x'y = x \oplus y$$

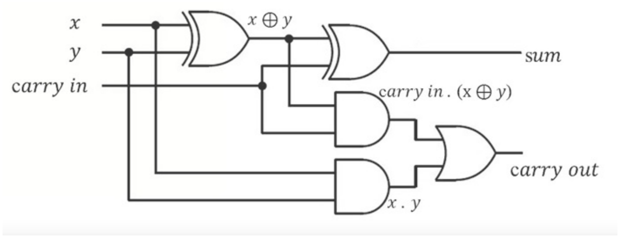
$$\text{carry} = xy$$



And a **full adder**:

$$\text{sum} = x \oplus y \quad \text{carry in}$$

$$\text{carry out} = xy + \text{carry in}. \,(x \oplus y)$$



2.1. Simplification of Circuits (example)

Let's consider the following boolean expression:

$$E = ((xy)'z)'((x' + z)(y' + z'))'$$

Using the **De Morgan's laws** and **involution**:

$$\begin{aligned} E &= ((xy)'' + z')((x' + z)' + (y' + z')') \\ &= (xy + z')((x''.\, z') + y''.\, z'') \\ &= (xy + z')(xz' + yz) \end{aligned}$$

Using the **distributive laws**:

$$E = xyxz' + xyyz + z'xz' + z'yz$$

Using **commutative, idempotent** and **complement** laws:

$$E = xyz' + xyz + xz' + 0$$

Using **absorption** law:

$$E = xyz + xz'$$

Last updated 2022-12-10 16:48:24 UTC