

# Digital electronics and Boolean Algebra

## Additional notes and exercises

August 15, 2012

### 1 Background

These notes contain additional information and exercises (not assessed) covering introductory material on digital electronics and boolean algebra. If you haven't looked at lecture notes #6 and are not familiar with all the digital gates associated truth tables then you should go and read that now.

Logic gates are the building blocks of digital electronics. They are ultimately made with transistor circuits, but in the study of digital electronics we can generally treat them as 'black boxes'. Combining multiple logic gates together in a circuit allows complex behaviors to be realized. When digital signals are used to represent binary numbers, circuits composed of logic gates can perform mathematical operations. Ultimately, the CPU in your computer does this on a massive scale.

The mathematical tool we use for analyzing logic gate circuits is boolean algebra. It is a very usefull tool, allowing us to simplify down complex circuits and to find how to make a gate of a particular type by combining gates of other types. Boolean algebra take a little practice to get used to but is ultimately no more difficult than elementary algebra.

## 2 The rules

We start with a series of identities. These should be reasonably obvious once you adjust to thinking of the  $\cdot$  and  $+$  operators in the boolean algebra context.

$$\begin{aligned}A \cdot A &= A \\A \cdot 1 &= A \\A \cdot 0 &= 0 \\A + A &= A \\A + 1 &= 1 \\A + 0 &= A\end{aligned}\tag{1}$$

Similarly, the rules for commutation, association and distribution apply to the symbols used in boolean algebra in the same way as they apply to the symbols for addition and multiplication, for example *commutation*:

$$\begin{aligned}A + B &= B + A \\A \cdot B &= B \cdot A\end{aligned}\tag{2}$$

*association*:

$$\begin{aligned}A + (B + C) &= (A + B) + C \\A \cdot (B \cdot C) &= (A \cdot B) \cdot C\end{aligned}\tag{3}$$

and *distribution*:

$$\begin{aligned}A + (B \cdot C) &= (A + B) \cdot (A + C) \\A \cdot (B + C) &= A \cdot B + A \cdot C\end{aligned}\tag{4}$$

*Redundancy* is an additional rule, which can be written in words: **In a Boolean expression containing a sum of products, a product that contains all the factors of another product is redundant.** For example:

$$A \cdot B + A \cdot B \cdot C + A \cdot B \cdot D = A \cdot B.\tag{5}$$

In this example, the last two *or* terms are made redundant by the first term. If  $A$  and  $B$  are true then the expression will be true regardless of the value of  $C$  or  $D$ .

*De Morgan's theorem* is very useful for simplifying expressions. It is often applied to remove a negation bar over a large number or terms. Formally it states:

$$A \cdot B \cdot C \dots N = \overline{\overline{A} + \overline{B} + \overline{C} + \dots \overline{N}}\tag{6}$$

or

$$A + B + C + \dots N = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C} \dots \overline{N}} \quad (7)$$

In practice to apply it the steps you go through are:

1. AND is changed to OR and OR is changed to AND
2. Each variable is replaced by its complement
3. The logical state of the entire expression is replaced by the complementary value.

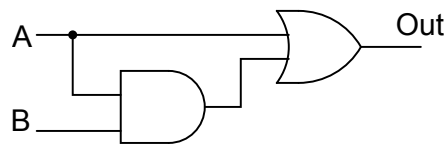
Note that a useful identity which follows from De Morgans theorem is

$$\overline{\overline{A}} = A \quad (8)$$

## 3 Examples

### 3.1 Example 1

Consider the circuit shown in Figure 1. Suppose we want to see if we can simplify this. An expression for the output can be written:



**Figure 1:** Circuit to simplify

$$A + (A \cdot B) \quad (9)$$

which we can begin to simplify. As  $A \cdot 1 = A$

$$(A \cdot 1) + (A \cdot B) \quad (10)$$

because from the distributive law (recall  $A \cdot B + A \cdot C = A \cdot (B + C)$ )

$$A \cdot (1 + B) \quad (11)$$

and  $B$  is redundant so we just have

$$A \cdot 1 = A \quad (12)$$

so in this example we don't actually need any gates in the circuit.

### 3.2 Example 2

Consider the following expression which we want to realize in a digital circuit:

$$\bar{C}(\overline{A+B}) + C(\bar{A} + \bar{B}) + A\bar{B}C + ABC \quad (13)$$

first we note that  $B$  is redundant in the last two terms

$$\bar{C}(\overline{A+B}) + C(\bar{A} + \bar{B}) + AC \quad (14)$$

to simplify the negated part of the first part of the expression we can apply De Morgans theorem.

$$\bar{C}(\bar{A} \cdot \bar{B}) + C(\bar{A} + \bar{B}) + AC \quad (15)$$

expanding

$$\bar{C}\bar{B}\bar{A} + C\bar{A} + C\bar{B} + AC \quad (16)$$

which can also be written

$$\bar{C}\bar{B}\bar{A} + C(\bar{A} + A) + C\bar{B} \quad (17)$$

the identity can be removed and the expression written

$$\bar{C}\bar{B}\bar{A} + (\bar{B} + 1)C \quad (18)$$

and since  $\bar{B} + 1 = 1$  we have

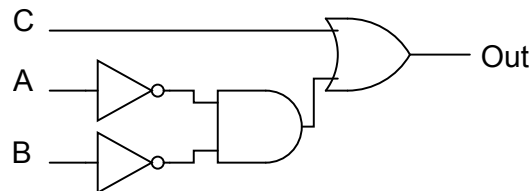
$$\bar{C}\bar{B}\bar{A} + C \quad (19)$$

and the  $\bar{C}$  term is redundant here so

$$\bar{A}\bar{B} + C \quad (20)$$

represents our final solution.

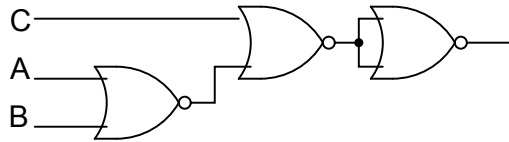
We could go ahead and implement this expression directly, as shown in Figure 2. However, in practice this is probably not an ideal solution. Gates tend to come packaged with several of the same type on the same chip and it is usually more important to minimize the number of chips than the number of logical gates.



**Figure 2:** Direct implementation of Equation 20

Suppose we only have NOR gates available. Applying De Morgan's theorem we can replace the left most gates with a single NOR gate. We can also make an OR gate out of one NOR gate followed by another configured as an inverter. This is shown in Figure 3.

Note that NOR and NAND gates are known as *universal gates* as it is possible to use combinations of either to represent any logical operation.



**Figure 3:** NOR implementation of Equation 20

## 4 Exercises

1. Convert the following decimal numbers into 8-bit binary representation : 12, 15, 65
2. Simplify  $A \cdot (B + A \cdot B)$
3. Simplify  $\bar{A} \cdot (B + A \cdot B)$
4. Simplify  $A + (B \cdot (\bar{A} + C) + C) + A \cdot B \cdot (\bar{D} + \bar{E})$
5. Simplify  $\overline{(A + B \cdot \bar{A}) + (C + D + E \cdot \bar{C})}$
6. Starting from a truth table for the XOR expression, show that  $A \oplus B = \bar{A} \cdot B + A \cdot \bar{B}$ . Try to represent this as a circuit containing only NAND gates.
7. Determine the Boolean algebra expression associated with the truth table below using each statement on the truth table that is TRUE. Minimize the resultant expression.

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