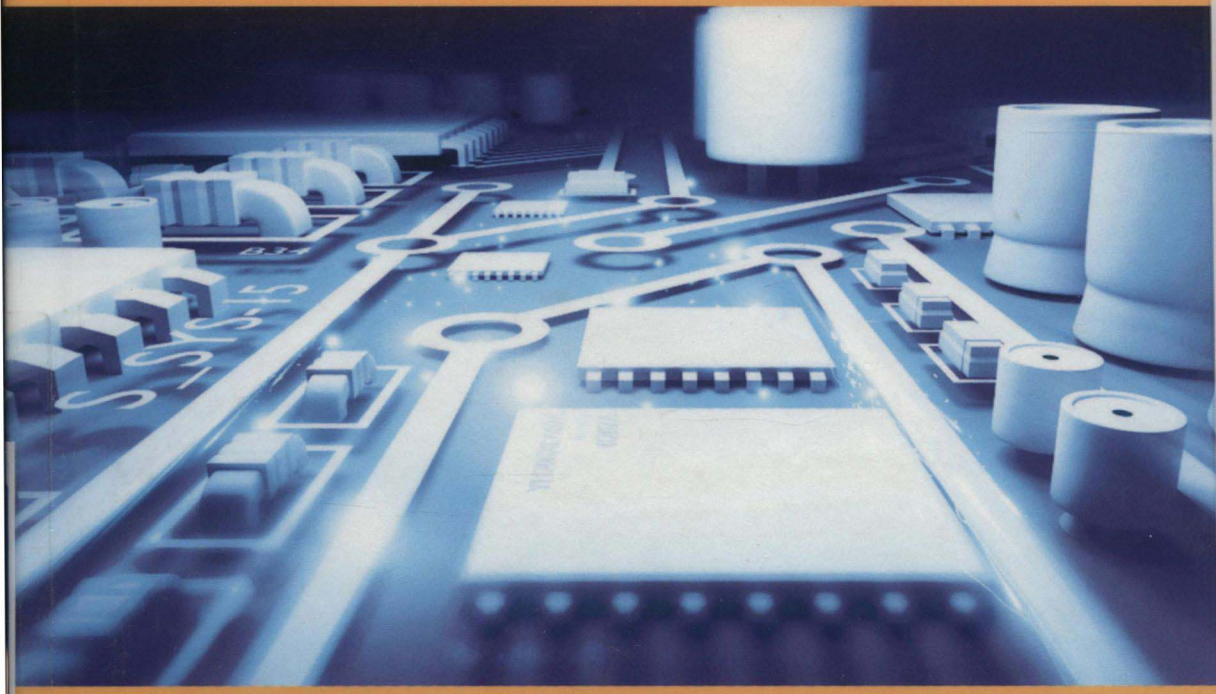
ELECTRONICS ENGINEERING SERIES

Ultra Low Power Electronics and Adiabatic Solutions

Hervé Fanet



ISTE

WILEY

Series Editor
Robert Baptist

Ultra Low Power Electronics and Adiabatic Solutions

Hervé Fanet

ISTE

WILEY

First published 2016 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

Library of Congress Control Number: 2016941915

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN 978-1-84821-738-6

Ultra Low Power Electronics and Adiabatic Solutions

Introduction

Energy efficiency is currently at the center of electronic and computer evolution. In fact, the objective of all three layers of information and communication technologies (i.e. high-performance servers and computers, mobile systems and connected objects) is to improve energy efficiency, meaning to compute more while consuming less. The costs of cooling systems' centers need to be restricted, the autonomy of portable systems needs to be increased and autonomous objects capable of functioning only on the energy that they recover need to be invented.

In these three cases, the power measurements are very different: kilowatts for servers, watts for mobile systems and micro-watts for connected objects. However, the mechanism that creates heat is the same in all the three cases and is due to the Joule effect. Two sources of dissipation have been identified: the first is the energy dissipated during the operations of charging and discharging the active electronic circuit capacitances and the second is the energy dissipated by currents that circulate permanently from the supply source to the ground when the circuits are in the sub-threshold regime. Therefore, it is necessary to fully understand these two phenomena in order to identify the causes that create heat and the possible paths for improvement. This is the objective of the first two chapters, which analyze the logic families. Thus, there appear to be links between heat creation and whether or not information is lost in logical operations. Chapter 3 provides the physical foundations necessary for understanding how the CMOS technology components in current use work.

Electronics has been confronting this crucial problem since the 2000s, as contrary to the initial predictions, it is no longer possible to pair the decrease

in transistor size with a decrease in supply voltage. Therefore, the density of the dissipated power does not stop growing in an integrated circuit. In Chapters 4 and 5, more and more sophisticated optimization techniques are described, which allow us to more or less restrict heat creation and energy consumption, but no solution seems to be capable of providing the long-awaited benefits. The analysis carried out in this book shows that for the current circuit architecture, the limit is intrinsic to semiconductor-based technologies, and that significant improvements can only be made by throwing the circuit architecture and component technology into question again. In order to achieve these objectives, new solutions (adiabatic computing and nano-relay technology) are proposed and described in Chapters 7 and 8. Chapter 5 is dedicated to reversible computing, considered by some to be the only solution for achieving extremely weak dissipation levels. It is also an introduction to quantum computing, which can be considered as an extension of reversible computing.

In summary, this book is an introduction to new possible directions in the evolution of electronic and computing systems. New directions will allow these systems to move beyond concepts that are dictated mainly by research on speed (which explains how electronics has evolved from the 1950s to the 2000s), to concepts that are inspired by the research of excellent energy efficiency.

Contents

1

Dissipation Sources in Electronic Circuits

This chapter explains the origins of how heat is created in electronic circuits, and details its two fundamental components: dynamic power and static power. Dynamic power is the heat that is produced by charging and discharging of the circuit capacitors when the logical states change, whereas static power is the heat that is dissipated by the Joule effect when there is current leakage, or when currents below the threshold circulate within the circuit's components. To fully understand how these mechanisms work, we need to analyze the different types of logical circuit structures. For this reason, we have dedicated a whole section to this subject. Logic based on complementary metal oxide semiconductor (CMOS) technology, which is used in more than 90% of current integrated circuits, will be explained in detail. The general principles put forward in this chapter will give the reader a fairly simple global view of the different aspects of heat production in circuits, and will allow them to understand the most important developments in semiconductor-based technology for reducing consumption. The more theoretical aspects will be discussed in Chapter 2 and the more detailed components of CMOS technology will also be discussed in Chapter 3.

1.1. Brief description of logic types

1.1.1. *Boolean logic*

In computer, audiovisual and control-command systems, data is binary-coded. This is true not only for the numbers, but also for the letters and, by extension, the sounds and images. Information processing systems perform the operations, from the simplest (addition) to the most complex (Fourier transformation). All of these are done by manipulating two symbols that are traditionally called "0" and "1". In control-command systems, decisons are

taken according to the value of logical functions, for example the value of "AND" when two simultaneous events occur. The mathematical model used in each case is Boolean algebra, invented by the Irish mathematician George Boole.

The simplest function is that of a single variable $f(A)$. Four different functions can be defined according to the possible values of a variable A, as shown in Figure 1.1.

A	*f(A)*
0	0
1	0

A	*f(A)*
0	1
1	1

A	*f(A)*
0	0
1	1

A	*f(A)*
0	1
1	0

Figure 1.1. *Boolean functions with one variable*

The third function is a copy of the variable, and the fourth is the inverter function, written as $\bar{A}$.

For two-input variables, the number of functions possible is the most important, as there are 2^4 possible functions, as shown in Figure 1.2.

A	*B*	*f1*	*f2*	*f3*	*f4*	*f5*	*f6*	*f7*	*f8*	*f9*	*f10*	*f11*	*f12*	*f13*	*f14*	*f15*	*f16*
0	**0**	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	**1**	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	**0**	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	**1**	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Figure 1.2. *Boolean functions with two variables*

The functions *f2*, *f7* and *f8* are very well known in electronics. They are, respectively, the AND, the exclusive OR and the OR functions. They are marked as:

– AND function: $A.B$

– Exclusive OR function: $A \oplus B$

– OR function: $A+B$

The symbols translate a certain analogy with decimal calculations. For example:

$$1 \cdot 0 = 0 \quad 1.1 = 1 \quad 1+0 = 0+1 = 1$$

However,

$$1+1 = 1$$

The point, which is the AND Boolean symbol, is often omitted to simplify the script.

All of the following are very easily executed using Boolean functions: binary arithmetic functions (used in current processors) and classical operations (addition, unsigned or floating point multiplication). For example, the addition of the bit i in Boolean algebra is written as S_i and C_i is the carry:

$$S_i = A_i \oplus B_i \oplus C_i \qquad [1.1]$$

$$C_{i+1} = A_i \cdot C_i + B_i \cdot C_i + A_i \cdot B_i \qquad [1.2]$$

We can now transpose the functions to the material level. The two states "0" and "1" are represented by physical quantities: two electrical voltages, for example. When the two voltages have two possible values 0 and V_{DD}, the same symbols "0" and "1" are assigned indifferently to the logical and physical values. We can talk about the two states in the same way: the "on" state and the "off" state. We also note that the logical states can be materialized by different physical quantities of the electrical voltage: for instance the magnetic moment or the polarization. When a logical function is materialized, it is called a logic gate.

Boolean algebra comprises a large number of rules that are shown in the Truth Tables for the functions in question. These rules allow us to simplify the logical expressions in the Truth Tables. It is no longer indispensable to memorize these techniques, as they are all now integrated into synthesis

tools. Let us make an exception for De Morgan's rules, which are often useful for understanding how data logic gates work:

$$\overline{A+B} = \overline{A} \cdot \overline{B} \qquad [1.3]$$

$$\overline{A \cdot B} = \overline{A} + \overline{B} \qquad [1.4]$$

The elementary demonstration is based on the Truth Tables.

The two Boolean function decompositions are called "Minterm" and "Maxterm", which are directly deducted from the Truth Tables. The simplest way to understand that is to use the example of figure 1.3, as it can serve as a generalization.

A	*B*	*C*	*f*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Figure 1.3. *Example of a three-variable function*

The Minterm decomposition is obtained by identifying the input values corresponding to a value of "1":

$$f(A,B,C) = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC$$

The Maxterm decomposition is obtained by reversing this to identify the input values corresponding to the value of "0" as an output:

$$f(A,B,C) = (A+B+C)(A+\overline{B}+\overline{C})(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+\overline{C})$$

Reed–Muller's decomposition is another decomposition that is fairly close to that of Taylor's series function. It is based on the two equations given below:

$$\overline{X} = 1 \oplus X$$

$$X + Y = X \oplus Y \oplus XY$$

Starting with the Minterm decomposition, it is possible to obtain an expression that only contains exclusive OR functions. Taking the example of the function given earlier, we obtain:

$$f(A,B,C) = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + AB\overline{C} = (1 \oplus A)(1 \oplus B)C + \ \ldots$$

In this case, after simplifying, we obtain:

$$f(A,B,C) = A \oplus B \oplus C \oplus AB$$

Generally, Reed–Muller's decomposition presents the function as a sum within the exclusive OR of input variable products:

$$\begin{aligned} f(A,B,C) = {} & c_{000} \oplus c_{100}A \oplus c_{010}B \oplus c_{001}C \oplus \\ & c_{110}\,AB \oplus c_{101}AC \oplus c_{011}BC \oplus c_{111}ABC \end{aligned} \qquad [1.5]$$

The factors are equal to 0 or 1.

To finish this introduction to Boolean algebra, let us introduce the notion of a Boolean function's partial derivative:

$$\frac{\partial f}{\partial A_i} = \frac{f(A_i = 1) \oplus f(A_i = 0)}{1 \oplus 0} \qquad [1.6]$$

This last notion, however, is not often used in the study of logical functions.

The decomposition of Boolean functions allows logical operations to materialize. Let us go back to the example given previously:

$$f(A,B,C) = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + AB\overline{C}$$

The basic "AND", "OR" and inverter functions are assumed to be carried out by material blocks, which can be combined in any way. In practice, however, this property is not always guaranteed and the most frequent occurrence is where an output can only be applied to a limited number of inputs. This is what is meant by "fan-out". The design of this simple function (Figure 1.4) shows the relative complexity of interconnect. This observation will be discussed in detail in the following section.

Knowing how many types of gates are necessary to carry out a particular function is a legitimate concern. The example given shows that the inverter, AND and OR functions are sufficient. In fact, we can dispense with the AND or OR functions by using De Morgan's laws. The inverse function and the AND gate form the complete basis, from which we are able to generate all the possible functions. It is the same for inverter and the OR gate. Gates with more than two inputs can be easily performed based on two-input gates, but it is more useful to perform these gates directly if the technology permits.

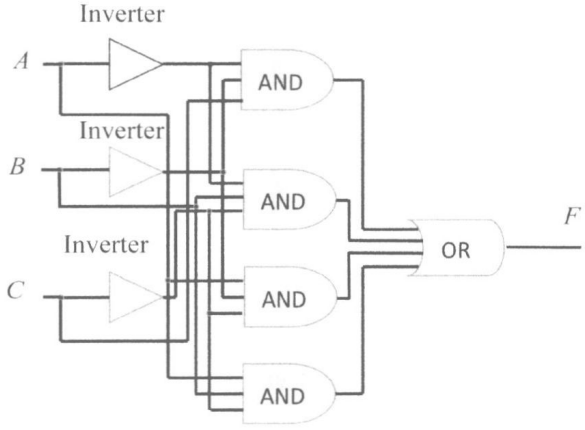


Figure 1.4. *Boolean material architecture*

To finish this brief introduction, we note that the NAND gate, that is to say the inverted AND, is enough on its own to generate all of the possible functions because if an input is permanently maintained at level "1", it brings about the inverter function.

The logical function synthesis is a technique that will not be explained in detail in this book. The aim of the synthesis is to create a circuit using as few gates as possible and to minimize the delay between the inputs and outputs.