TURING 图灵原版电子与电气工程系列

ELSEVIER 爱思唯尔

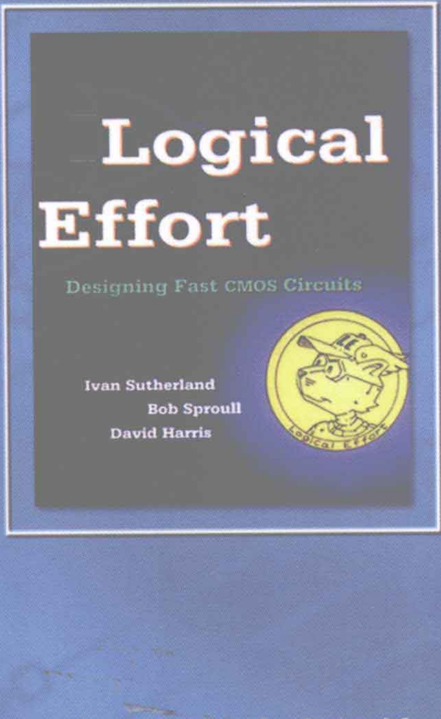
Logical Effort
Designing Fast CMOS Circuits

高速CMOS电路设计
Logical Effort方法

（英文版）

[美] Ivan Sutherland
Bob Sproull
David Harris 著

图灵奖得主著作
IC设计师秘籍


Logical Effort
Designing Fast CMOS Circuits
Ivan Sutherland
Bob Sproull
David Harris

人民邮电出版社
POSTS & TELECOM PRESS

Logical Effort: Designing Fast CMOS Circuits

高速CMOS电路设计 Logical Effort方法（英文版）

为了满足速度需求，集成电路设计师常常要痛苦地在无数选择中反复调整自己的设计，费时费力。两位计算机科学大师针对这一问题提出了一种简单而普遍有效的方法：Logical Effort。本书就是他们对这一方法全面而生动的阐述。

通过本书，你不仅能够迅速地理解和掌握Logical Effort方法，大大提高自己的工作效率，而且还能从大师著作的字里行间领悟到更多思想精髓。

Ivan Sutherland 著名计算机科学家。因对计算机图形学和电子设计领域的开创性贡献先后获得1988年图灵奖和1998年冯·诺依曼奖。美国科学院院士、美国工程院院士和ACM会士。现任Sun公司副总裁。

Bob Sproull 著名计算机科学家，美国工程院院士。现为Sun公司副总裁兼研究中心主任。Sutherland的长期合作者。

David Harris Harvey Mudd学院副教授。曾参与Intel安腾和奔腾II的电路设计。除本书外，他还与Weste合著了名作*CMOS VLSI Design: A Circuits and Systems Perspective*。

ELSEVIER
爱思唯尔

本书相关信息请访问：图灵网站 http://www.turingbook.com
读者/作者热线：(010) 88593802
反馈/投稿/推荐信箱：contact@turingbook.com

分类建议 电子电气/电子设计

人民邮电出版社网址 www.ptpress.com.cn

ISBN 978-7-115-19598-2

9 787115 195982 >

ISBN 978-7-115-19598-2/TN

定价：45.00 元

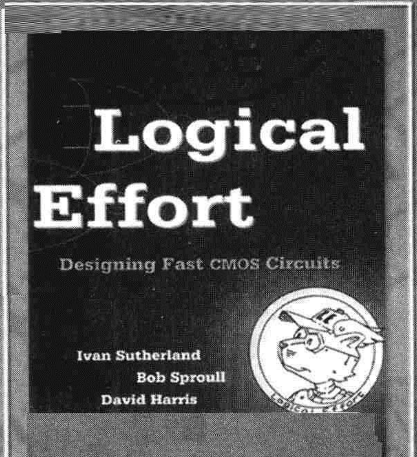
TURING 图灵原版电子与电气工程系列

Logical Effort
Designing Fast CMOS Circuits

高速CMOS电路设计
Logical Effort方法

（英文版）

Ivan Sutherland
[美] Bob Sproull 著
David Harris



人民邮电出版社
北　京

图书在版编目（CIP）数据

高速 CMOS 电路设计：Logical Effort 方法=Logical Effort: Designing Fast CMOS Circuits：英文/（美）萨瑟兰德（Sutherland, I.），（美）斯普劳尔（Sproull, B.），（美）哈里斯（Harris,D.）著.—北京：人民邮电出版社，2009.3
(图灵原版电子与电气工程系列)
ISBN 978-7-115-19598-2

Ⅰ. 高… Ⅱ. ①萨…②斯…③哈…Ⅲ. 互补 MOS 集成电路—电路设计—英文 Ⅳ. TN432.02

中国版本图书馆 CIP 数据核字（2008）第 212395 号

内 容 提 要

本书讲述如何获得高速 CMOS 电路，这正是高速集成电路设计师们渴望获得的技术。在设计中，我们往往面对无数的选择，本书将告诉我们如何将这些选择变得更容易和更有技巧。本书提供了一个简单而普遍有效的方法，用于估计拓扑、电容等因素造成的延迟。

本书实用性强，适合集成电路设计师以及相关专业的师生。

图灵原版电子与电气工程系列

高速 CMOS 电路设计：Logical Effort 方法（英文版）

◆ 著　　[美] Ivan Sutherland Bob Sproull David Harris
　责任编辑　舒　立
◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
　邮编　100061　　电子函件　315@ptpress.com.cn
　网址　http://www.ptpress.com.cn
　北京艺辉印刷有限公司印刷
◆ 开本：787×1 092　1/16
　印张：16
　字数：256 千字　　　2009 年 3 月第 1 版
　印数：1－2 000 册　　2009 年 3 月北京第 1 次印刷

著作权合同登记号　图字：01-2008-5823 号

ISBN 978-7-115-19598-2/TN

定价：45.00 元

读者服务热线：(010)88593802　印装质量热线：(010)67129223
反盗版热线：(010) 67171154

How It All Started

The method of logical effort evolved in three stages. It began in 1985 while I was living in London. Bob Sproull and I were engaged in research on fast asynchronous circuits involving mostly Muller C-elements and XOR functions. In trying to improve the speed of our circuits I resorted to calculus for lack of circuit simulation tools. Instead of computing I had to think about the problem, a formula for success that I recommend highly. Fortunately, both Muller C-elements and XOR functions are symmetric with respect to zero and one, and the usual circuits for them are correspondingly symmetric in N and P transistors. Their delay equations revealed a simple similarity between logic delay and electrical delay. It was only later that we learned to treat less symmetric functions like NAND and NOR.

I recall well the period of about a week during which the idea of logical effort emerged. At first I had only hints that the equations were telling me something interesting; I could smell value before simplicity emerged. I wrote a memo to Bob Sproull trying to describe the concept, but the formulation was still unclear and the idea had no name.

With more understanding I was able to name the idea "logical effort." Logical effort described the increased cost inherent in the circuit topology necessary to implement a logic function. I was pleased that more complex logic functions had higher logical effort than simple ones, and that the logical effort of compound circuits was the product of their individual logical efforts. With the name logical effort assigned and precisely defined, the idea became useable. The name electrical effort came afterward, assigning a name to a problem whose solution—namely gain—was very well understood.

The second phase of evolution took place in the late 1980s. I had returned to the United States, and worked with Bob Sproull to prepare the class notes on which this text is based. Bob carried the mathematics further than I had, finding ways to deal with parasitic delays that I had previously ignored. He tidied our notation, fixed my prose, and augmented my rough notes so that we could teach a coherent course to our industrial sponsors. We had almost a book but lacked the energy to finish it. In 1991 Bob and I published a short paper about logical effort [8].

Years later, David Harris faced the problem of teaching junior circuit designers and graduate students at Stanford University how to design circuits and size transistors. Teaching is often the best way to learn; he was forced to develop coherent explanations for his intuitive approach to sizing. His explanations proved to be a rediscovery of logical effort, which suggests that logical effort may be fundamental to circuit topology. David gradually discovered more properties of circuits, especially regarding the logical effort of newer circuit families such as domino logic. When David and I met, we found that many of his results were already in the unpublished logical effort course notes. Because he and his students wanted a good reference text for logical effort, David undertook the task of polishing the course notes into book form. Youth has such energy.

Ivan Sutherland

Preface

The method of logical effort is a way of thinking about delay in MOS circuits. It seeks to determine quickly a circuit's maximum possible speed and how to achieve it. It provides insight into how both the sizes of different transistors and the circuit topology itself affect circuit delay.

We offer two new names for causes of delay in MOS circuits, *electrical effort* and *logical effort*. The similarity of these names reflects a remarkable symmetry between the effort required to drive an electrical load and the effort required to perform a logic function; the two forms of effort present identical and interchangeable sources of delay. Identifying these concepts leads to a formulation that simplifies circuit analysis and allows a designer to analyze alternative circuit designs quickly.

Electrical effort is a new name for the problem overcome by electrical gain. It has long been known that the fastest driver for a large electrical load is a multistage amplifier whose gain is distributed among stages of exponentially increasing size. Thinking of what amplifiers do as compensating for electrical effort paves the way to understanding how they similarly compensate for logical effort.

Logical effort describes the cost of computation inherent in the circuit topology that implements each logic function. Logic functions incur a cost not only because they involve many transistors, but also because MOS transistors in series are poorer conductors of electricity than individual transistors of the same size. Both factors conspire to make logic function blocks less good than inverters at electrical amplification. Logical effort quantifies this weakness, enabling us to reason about which of several alternate topologies will be best.

Critics of this method observe that it achieves no more than conventional RC analysis and that experienced designers know how to optimize circuits for speed. Indeed, the best designers, whether by intuition or experience, design circuits that match closely those derived by the method of logical effort. However, we have seen many instances where experienced designers devise poor circuits. Even the best designers can become mired in detailed transistor sizing simulations and fail to find structural changes to a circuit that will lead to major performance improvements. Because of its simplicity, the method of logical effort bridges the gap between structural design and detailed simulation.

We wrote this book for those who design MOS integrated circuits. It assumes a knowledge of static CMOS digital circuits, elementary electronics, and modest mathematical skill. Although some of the derivations use calculus, only algebra is required to apply the method. The novice designer will find simple techniques for designing high-speed circuits. The experienced designer will find new ways to think about old design techniques. Both will gain new rules of thumb that lead to high-speed circuits. The techniques of logical effort help us analyze and optimize large circuits quickly.

How to Use this Book

There are many ways to use this text. We believe it will be of interest to practicing circuit, logic, and CAD designers, students, and researchers. Junior circuit designers will learn new techniques and reduce their dependence on tedious circuit simulation, while veteran designers will discover new ways to look at concepts they may have developed intuitively through experience. We believe that logic designers interested in high-speed chips must have a thorough understanding of delay in CMOS gates. Logical effort provides simple but powerful models for thinking about this delay and comparing alternative topologies. Similarly, we believe good tool developers need a thorough understanding of the problems being faced by their users, and we hope this book will offer them such insight.

Chapter 1 stands alone as an introduction to logical effort. A road map at the end of the chapter describes the more advanced topics presented later in the book. A course on VLSI design may use the first four chapters as supplemental reading to provide examples of applying logical effort and to develop the basic theory behind the method. Experienced circuit designers and students in advanced circuit classes will be interested in the later chapters, which apply logical effort to common circuit problems. We conclude with Chapter 12, a concise review of the method of logical effort and of important insights gained from the method.

About the Exercises

In our experience, it is very difficult to learn anything without practice. We have provided a number of exercises at the end of each chapter intended for self-study as well as for formal classes in logical effort.

The problems are rated in difficulty on a logarithmic scale, similar to that used by Knuth and Hennessy. A rough guide is listed below. Your mileage may vary.

[10]	1 minute (read and understand)
[20]	15–20 minutes
[30]	2 hours or more (especially if the TV is on)
[50]	research problem

Solutions to the odd-numbered problems are presented in the back of the book. Please use them wisely; do not turn to the answer until you have made a genuine effort to solve the problem yourself. Solutions to the even-numbered problems are available to instructors on the logical effort Web page (see the following section).

About the Web Site

The Morgan Kaufmann Web page *www.mkp.com/Logical_Effort* is dedicated to and offers several tools to assist with logical effort.

Some features on this Web page include

- A detailed example of logical effort applied to the design of a multiplier.
- Solutions to even-numbered exercises, available to instructors.
- The Perl script used in Chapter 5 to characterize the logical effort of gates. The script takes a SPICE netlist of the gates, a process file, and a list of input

stimuli for each gate. It measures the logical effort and parasitic delay of each gate using the test setup described in Chapter 5.

- A Java tool to design wide NAND, NOR, AND, and OR gates. It takes the number of inputs and the electrical effort of the path and computes the minimum-delay tree, as discussed in Section 11.1. This tool can be used from a form-based interface on the Web, or downloaded for use on your computer.

If you discover an error in this book, please contact the publisher by email at lebugs@mkp.com. The first person to report a technical error will be awarded a $1.00 bounty upon its implementation in future printings of the book. Please check the errata page at *www.mkp.com/Logical_Effort* to see if a particular bug has already been reported and corrected.

Acknowledgments

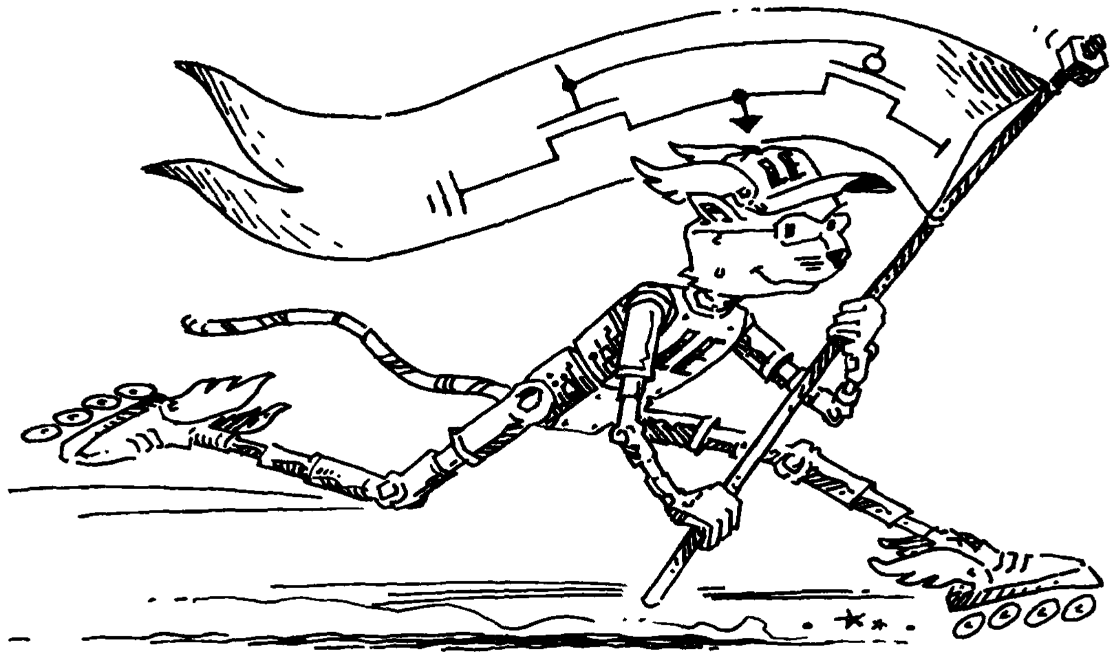
Many people have helped us develop the method of logical effort and prepare this book. We wish to thank five companies that sponsored the original research: Austek Microsystems, Digital Equipment Corporation, Evans and Sutherland Computer Corporation, Floating Point Systems, and Schlumberger. We are grateful to Apple Computer for its support as we began to edit our course notes. We also thank the engineers and designers from those firms who served as students during our early attempts to teach this material and whose penetrating questions contributed to a clearer presentation of the ideas. We thank Carnegie Mellon University, Stanford University, and the Imperial College of London University for the office space, computing support, and collegial thinking they have provided.

Thanks are due to Sun Microsystems Laboratories for encouragement and support as we brought this book into final form, Our colleagues Ian W. Jones, Erik L. Brunvand, Bob Proebsting, Mark Horowitz, and Peter Single contributed in several ways to the work. More recently, we thank our students at Stanford University, HAL Computer, UC Berkeley, and Intel Corporation for bringing fresh life and interest to logical effort. Thanks are also due to those who participated in the review process: Peter Ashenden of the University of Adelaide, Peter Beerel and Massoud Pedram of the University of Southern California, Dileep Bhandarkar of Intel, Lynn Conway of the University of Michigan, Steve Kang and Farid Najm of the University of Illinois, Urbana-Champaign, Jaeha Kim of Stanford University, and Wayne Wolf of Princeton University.

We have been very happy working with our publisher, Morgan Kaufmann. We would especially like to thank our editor, Denise Penrose, editorial coordinator, Meghan Keeffe, and production editor, Edward Wade, for their devotion to quality and good sense of humor.

Jaeha Kim did a remarkably thorough job hunting down errors in the text and solutions; the remaining mistakes are ours. Sally Harris worked tirelessly to prepare illustrations. Finally, we offer special thanks to our friends and colleagues Bob Spence and the late Charles Molnar for ideas, encouragement, and moral support.

Contents



The Method of Logical Effort——1

Designing a circuit to achieve the greatest speed or to meet a delay constraint presents a bewildering array of choices. Which of several circuits that produce the same logic function will be fastest? How large should a logic gate's transistors be to achieve least delay? And how many stages of logic should be used to obtain least delay? Sometimes, adding stages to a path reduces its delay!

The *method of logical effort* is an easy way to estimate delay in a CMOS circuit. We can select the fastest candidate by comparing delay estimates of different logic structures. The method also specifies the proper number of logic stages on a path and the best transistor sizes for the logic gates. Because the method is easy to use, it is ideal for evaluating alternatives in the early stages of a design and provides a good starting point for more intricate optimizations.

This chapter describes the method of logical effort and applies it to simple examples. Chapter 2 explores more complex examples. These two chapters together provide all you need to know to apply the method of logical effort to a wide class of circuits. We devote the remainder of this book to derivations that show why the method of logical effort works, to some detailed optimization techniques, and to the analysis of special circuits such as domino logic and multiplexers.

1.1 —— Introduction

To set the context of the problems addressed by logical effort, we begin by reviewing a simple integrated circuit design flow. We will see that topology selection and gate sizing are key steps of the flow. Without a systematic approach, these steps are extremely tedious and time-consuming. Logical effort offers such an approach to these problems.

Figure 1.1 shows a simplified chip design flow illustrating the logic, circuit, and physical design stages. The design starts with a specification, typically in textual form, defining the functionality and performance targets of the chip. Most chips are partitioned into more manageable blocks so that they may be divided among multiple designers and analyzed in pieces by CAD tools. Logic designers write register transfer level (RTL) descriptions of each block in a language like Verilog or VHDL and simulate these models until they are convinced the specification is correct. Based on the complexity of the RTL descriptions, the designers estimate the size of each block and create a floorplan showing relative placement of the blocks. The floorplan allows wire-length estimates and provides goals for the physical design.

Given the RTL and floorplan, circuit design may begin. There are two general styles of circuit design: custom and automatic. *Custom* design trades additional human labor for better performance. In a custom methodology, the circuit designer has flexibility to create cells at a transistor level or choose from a library of predefined cells. The designer must make many decisions: Should I use static CMOS, transmission gate logic, domino circuits, or other circuit families? What circuit topology best implements the functions specified in the RTL? Should I use NAND, NOR, or complex gates? After selecting a topology and drawing the schematics, the designer must choose the size of transistors in each logic gate. A larger gate drives its load more quickly, but presents greater input capacitance to the previous stage and consumes more area and power. When the schematics are complete, functional verification checks that the schematics correctly implement the RTL specification. Finally, timing verification checks that the circuits meet the performance targets. If performance is inadequate, the circuit designer may try to resize gates for improved speed, or may have to