JANUSZ A. BRZOZOWSKI
MICHAEL YOELI

Digital Networks

PRENTICE-HALL
SERIES IN
AUTOMATIC
COMPUTATION

DIGITAL NETWORKS

JANUSZ A. BRZOZOWSKI

University of Waterloo

MICHAEL YOELI

Technion—Israel Institute of Technology

PRENTICE-HALL, INC.

ENGLEWOOD CLIFFS, N.J.

Library of Congress Cataloging in Publication Data

BRZOZOWSKI, J A
Digital networks.

(Automatic computations)
Bibliography: p.
Includes index.
1. Digital electronics. I. Yoeli, M. (date)-
joint author. II. Title.
TK7868.D5B78 621.3815'3 75-17966
ISBN 0-13-214189-2

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA, PTY. LTD., *Sydney*
PRENTICE-HALL OF CANADA, LTD., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA (PTE.) LTD., *Singapore*

DIGITAL NETWORKS

Prentice-Hall
Series in Automatic Computation

AHO, ed., *Currents in the Theory of Computing*
AHO AND ULLMAN, *The Theory of Parsing, Translation, and Compiling*, Volume I: *Parsing*; Volume II: *Compiling*
ANDREE, *Computer Programming: Techniques, Analysis, and Mathematics*
ANSELONE, *Collectively Compact Operator Approximation Theory and Applications to Integral Equations*
BATES AND DOUGLAS, *Programming Language/One*, 2nd ed.
BLUMENTHAL, *Management Information Systems*
BRENT, *Algorithms for Minimization without Derivatives*
BRINCH HANSEN, *Operating System Principles*
BRZOZOWSKI AND YOELI, *Digital Networks*
COFFMAN AND DENNING, *Operating Systems Theory*
CRESS, et al., *FORTRAN IV with WATFOR and WATFIV*
DAHLQUIST, BJORCK, AND ANDERSON, *Numerical Methods*
DANIEL, *The Approximate Minimization of Functionals*
DEO, *Graph Theory with Applications to Engineering and Computer Science*
DESMONDE, *Computers and Their Uses*, 2nd ed.
DRUMMOND, *Evaluation and Measurement Techniques for Digital Computer Systems*
ECKHOUSE, *Minicomputer Systems: Organization and Programming (PDP-11)*
FIKE, *Computer Evaluation of Mathematical Functions*
FIKE, *PL/I for Scientific Programmers*
FORSYTHE AND MOLER, *Computer Solution of Linear Algebraic Systems*
GEAR, *Numerical Initial Value Problems in Ordinary Differential Equations*
GORDON, *System Simulation*
GRISWOLD, *String and List Processing in SNOBOL4: Techniques and Applications*
HANSEN, *A Table of Series and Products*
HARTMANIS AND STEARNS, *Algebraic Structure Theory of Sequential Machines*
JACOBY, et al., *Iterative Methods for Nonlinear Optimization Problems*
JOHNSON, *System Structure in Data, Programs, and Computers*
KIVIAT, et al., *The SIMSCRIPT II Programming Language*
LAWSON AND HANSON, *Solving Least Squares Problems*
LORIN, *Parallelism in Hardware and Software: Real and Apparent Concurrency*
LOUDEN AND LEDIN, *Programming the IBM 1130*, 2nd ed.
MARTIN, *Computer Data-Base Organization*
MARTIN, *Design of Man-Computer Dialogues*
MARTIN, *Design of Real-Time Computer Systems*
MARTIN, *Future Developments in Telecommunications*
MARTIN, *Programming Real-Time Computing Systems*
MARTIN, *Security, Accuracy, and Privacy in Computer Systems*
MARTIN, *Systems Analysis for Data Transmission*
MARTIN, *Telecommunications and the Computer*

MARTIN, *Teleprocessing Network Organization*
MARTIN AND NORMAN, *The Computerized Society*
MCKEEMAN, et al., *A Compiler Generator*
MYERS, *Time-Sharing Computation in the Social Sciences*
MINSKY, *Computation: Finite and Infinite Machines*
NIEVERGELT, et al., *Computer Approaches to Mathematical Problems*
PLANE AND MCMILLAN, *Discrete Optimization: Integer Programming and Network Analysis for Management Decisions*
POLIVKA AND PAKIN, *APL: The Language and Its Usage*
PRITSKER AND KIVIAT, *Simulation with GASP II: A FORTRAN-based Simulation Language*
PYLYSHYN, ed., *Perspectives on the Computer Revolution*
RICH, *Internal Sorting Methods Illustrated with PL/I Programs*
SACKMAN AND CITRENBAUM, eds., *On-Line Planning: Towards Creative Problem-Solving*
SALTON, ed., *The SMART Retrieval System: Experiments in Automatic Document Processing*
SAMMET, *Programming Languages: History and Fundamentals*
SCHAEFER, *A Mathematical Theory of Global Program Optimization*
SCHULTZ, *Spline Analysis*
SCHWARZ, et al., *Numerical Analysis of Symmetric Matrices*
SHAH, *Engineering Simulation Using Small Scientific Computers*
SHAW, *The Logical Design of Operating Systems*
SHERMAN, *Techniques in Computer Programming*
SIMON AND SIKLOSSY, eds., *Representation and Meaning: Experiments with Information Processing Systems*
STERBENZ, *Floating-Point Computation*
STOUTEMYER, *PL/1 Programming for Engineering and Science*
STRANG AND FIX, *An Analysis of the Finite Element Method*
STROUD, *Approximate Calculation of Multiple Integrals*
TANENBAUM, *Structured Computer Organization*
TAVISS, ed., *The Computer Impact*
UHR, *Pattern Recognition, Learning, and Thought: Computer-Programmed Models of Higher Mental Processes*
VAN TASSEL, *Computer Security Management*
VARGA, *Matrix Iterative Analysis*
WAITE, *Implementing Software for Non-Numeric Application*
WILKINSON, *Rounding Errors in Algebraic Processes*
WIRTH, *Algorithms + Data Structures = Programs*
WIRTH, *Systematic Programming: An Introduction*
YEH, ed., *Applied Computation Theory: Analysis, Design, Modeling*

To Grażyna and Nehama

PREFACE

Digital network engineering has developed very rapidly in recent years. Its impact is felt in many areas, especially those of digital computers, telephone switching, and digital control. The growth of digital systems engineering has been accompanied by the birth of new technologies, such as integrated circuits, large-scale integration, and MOS devices. This book is intended to be a realistic, up-to-date introduction to the theory and applications of digital networks.

For the application-oriented reader we provide a comprehensive and mathematically precise treatment of commercially available integrated-circuit modules, which have become the building blocks of modern digital systems. We introduce new mathematical models to properly explain the behavior of complex flip-flops (including master–slave and edge-sensitive types); and static MOS devices. We also make an effort toward establishing a modular design approach for large digital systems. Our presentation is at the logical design level and does not rely on electrical engineering prerequisites.

For the more theoretically inclined reader we provide mathematically nontrivial topics which are at the same time realistic and applicable. A considerable amount of the material in this book represents recent, unpublished research results. The reader will come across several problem areas that require further investigation.

Conventional switching theory originated with relay technology, but it did not keep up to date with the rapid developments of new technologies. Rather, it deals mainly with mathematical problems that presently have little relevance to actual practice in digital network design. On the other hand, a number of texts addressed to the practicing technician and engineer are not on the proper mathematical level for university undergraduates. In contrast to both these trends, we have strived for both relevance and mathematical maturity.

This book is addressed to more than one type of reader. Most of the material is suitable for an undergraduate, junior-level course for computer science and electrical engineering students. For example, the topics of Chapters 1, 2, 4 (unstarred part), 5, 6, 7, and 8 have been covered in a one-semester course for third-year computer science students at the University of Waterloo. Alternative sequences are also possible at the undergraduate level, for example Chapters 1, 2, 4 (unstarred part), 6, 7, 8, 9 (possibly with some parts omitted), and 10 (unstarred part). The first part of each chapter gives an indication of prerequisite material and points out sections that can be omitted. For a graduate course in computer science or electrical engineering, Chapters 3 through 10 constitute a suitable sequence.

Appendices A–C, together with Chapters 3 and 4, provide a good deal of general mathematical background required by computer science students.

We wish to express our gratitude for helpful comments, reactions, and criticisms of early versions of this book to the following: students at the University of Waterloo, especially G. Bouwers, T. A. Cargill, D. R. Cheriton, and M. G. Gouda; Dr. R. Knast; Professors A. Bar-Lev, I. Kidron, W. D. Little, and J. C. Majithia; and several anonymous, very helpful reviewers.

Special thanks are due to M. I. Irland, who shared the experience of teaching from our manuscripts, for constructive criticism and for contributing to the problem sections.

Both authors wish to gratefully acknowledge the assistance of both the University of Waterloo and the Technion—Israel Institute of Technology.

The research for this book was supported in part by the National Research Council of Canada under Grant A-1617, and by the Technion, under Grant 120-509.

We are greatly indebted to Teresa Miao of the University of Waterloo for her excellent typing of the various stages of the manuscript, and to Raya Anavi of the Technion, for typing extensive revisions.

J. A. BRZOZOWSKI
M. YOELI

DIGITAL NETWORKS

CONTENTS

PREFACE xiii

1 SWITCHES AND GATES 1

About This Chapter 1
1.1 Notation 2
1.2 Switches 3
1.3 Gates 6
1.4 Boolean Operators 8
1.5 Gate Networks 12
1.6 Design Examples 14
Problems 21

2 COMBINATIONAL INTEGRATED CIRCUIT MODULES 25

About This Chapter 25
2.1 Combinational Networks 26
2.2 Integrated Circuits 28
2.3 Small-Scale Integration 30
2.4 Comparators 33
2.5 Selectors and Decoders 36
2.6 Priority Encoders 38
2.7 Code Converters 39
2.8 Static Read-Only Memories 45
2.9 Arithmetic-Logic Unit 47
Problems 55

3 BOOLEAN ALGEBRAS 57

About This Chapter 57
*3.1 Introduction 58
3.2 Boolean Algebras 58
*3.3 Posets 60
*3.4 Semilattices 64
*3.5 Lattices 66
*3.6 Boolean Lattices 74
*3.7 Characterizations of Boolean Algebras 76
Problems 82

4 BOOLEAN FUNCTIONS AND EXPRESSIONS 86

About This Chapter 86
4.1 Introduction 86
4.2 Boolean Functions 87
4.3 Boolean Expressions 91
4.4 Analysis of Combinational Networks 96
*4.5 Ternary Functions: More About Transient Phenomena 101
4.6 Summary of Algebraic Concepts 103
*4.7 Special Properties of Boolean Functions 106
Problems 118

5 SIMPLIFICATION OF BOOLEAN EXPRESSIONS 121

About This Chapter 121
5.1 Introduction 122
5.2 Simplification of Two-Level Expressions 122
5.3 Prime Implicants 125
5.4 Irredundant Sums 137
5.5 Partial Boolean Functions 141
5.6 Minimal Sums 144
5.7 Other Aspects of Minimization 146
5.8 Design Examples 148
Problems 160

6 INTRODUCTION TO SEQUENTIAL NETWORKS 164

About This Chapter 164
6.1 Gate Networks with Propagation Delays 165
6.2 Combinational Networks Again 167
6.3 Sequential Networks: Introductory Examples 172

6.4 Difficulties with the Ideal-Delay Model 179
6.5 Mathematical Models of Sequential Networks 182
6.6 Summarizing Comments 190
Problems 191

7 LATCHES AND FLIP-FLOPS 194

About This Chapter 194
7.1 NOR and NAND Latches 195
7.2 Simple One-Latch Flip-Flops 198
7.3 Master–Slave Flip-Flops 205
7.4 Edge-Sensitive Flip-Flops 218
7.5 Other Flip-Flops 229
7.6 Monostable Multivibrators 232
Problems 234

8 MORE COMPLEX SEQUENTIAL NETWORKS 240

About This Chapter 240
8.1 Ripple Counters 241
8.2 Shift Registers 245
8.3 Systematic Analysis of Synchronous-Mode Networks 248
8.4 Synthesis of Synchronous-Mode Sequential Networks 261
8.5 Iterative Networks 274
8.6 Modular Design of Synchronous Networks 275
8.7 IC Memories 279
Problems 288

9 DESIGN OF DIGITAL NETWORKS 291

About This Chapter 291
9.1 Digital Automatic Gain Controller 292
9.2 Service Request Controllers 299
9.3 Sequence Controllers 311
9.4 Implementation of Numerical Algorithms 322
Problems 327

10 ERRORS AND HAZARDS 330

About This Chapter 330
10.1 Error Detection in Digital Networks 330
10.2 Error Correction 342
*10.3 Theory of Logic Hazards 345
*10.4 Applications of Hazard Theory 351
Problems 354

APPENDICES

*A PROPOSITIONAL CALCULUS 355

B SETS 359

*C FUNCTIONS AND RELATIONS 365

D MODELS OF STATIC MOS GATES 370

REFERENCES 381

INDEX 385

1 SWITCHES AND GATES

ABOUT THIS CHAPTER

This chapter introduces switches, gates, and relatively simple gate networks. The approach is largely intuitive, yet it is made quite precise through the use of the notation of propositional calculus, given in Section 1.1. For the mathematically oriented reader, we present in Appendix A additional material on propositional calculus.

In Section 1.2 we begin with switches, because their implementation and operation are extremely simple conceptually. In Section 1.3 we introduce ideal gates without attempting to explain their numerous and widely differing implementations. A mathematical model of one type of implementation, MOS gates, is presented in Appendix D; however, we consider electronic details outside the scope of our book and refer the interested reader to the literature [GAR]. Thus we restrict our attention to logical properties of gates and we consider gates as operators on binary signals.

In Section 1.4, Boolean operators are defined and some of their properties are described; many of these properties are needed in Section 1.6. In Section 1.5, some fundamental concepts related to networks of gates are discussed.

With this rather modest background we next proceed to the design examples of Section 1.6. With the aid of the propositional-calculus notation, we develop an approach to the design of some interesting networks. Word descriptions of problems are first converted to precise mathematical statements and then manipulated to obtain gate implementa-

tions. For the most part we use ideal gates; however, certain restrictions are introduced in some of the examples. For instance, we attempt to use as few gates or contacts as possible. In this connection, the reader is warned that minimal solutions are quite difficult to find for some of the examples and problems. Such problems constitute intellectually challenging puzzles, but, otherwise, the reader should not attach too much importance to absolutely minimal solutions. In general, no systematic methods are known for finding such minimal networks. The subject is treated in more detail in Chapter 5.

To the best of our knowledge, the approach of Section 1.6 is original, although countless designers have undoubtedly gone through similar steps informally.

1.1. NOTATION

Throughout this book we frequently use notation from the calculus of propositions. We now summarize this notation. For a more detailed account of the propositional calculus, the reader is referred to Appendix A.

A *proposition* is a statement that is either true or false. If p and q are propositions, the proposition $p \wedge q$ (read: p AND q) is defined to be true iff (if and only if) both p *and* q are true. The proposition $p \vee q$ (read: p OR q) is true iff either p *or* q *or* both p and q are true. The proposition $\sim p$ (read: NOT p) is true iff p is *not* true, i.e., iff p is false. We write LHS $\equiv$ RHS to state that the left-hand side and right-hand side are *equivalent*; i.e., they are either both true or are both false. For example, $x > y \equiv y < x$. Another example of an equivalence is the following:

$$p \vee q \equiv p \vee ((\sim p) \wedge q), \qquad (*)$$

where p and q denote arbitrary propositions. This and similar equivalences can be verified by means of a *truth table*, which exhausts all the possibilities for p and q, as shown in Table 1-1, where T and F stand for *true* and *false*, respectively.

Table 1-1 Truth-Table Verification of Equivalence (*)

p	q	$p \vee q$	$\sim p$	$(\sim p) \wedge q$	$p \vee ((\sim p) \wedge q)$
F	F	F	T	F	F
F	T	T	T	T	T
T	F	T	F	F	T
T	T	T	F	F	T

We refer to the symbols $\wedge$, $\vee$, and $\sim$ as *logical connectives.*

For convenience in such formulas as $((f \vee g) \vee h)$, we omit the outer parentheses and write $(f \vee g) \vee h$. Also, the $\sim$ operator has precedence over the $\vee$ and $\wedge$ operators. Thus $p \vee ((\sim p) \wedge q)$ will be written $p \vee (\sim p \wedge q)$.

1.2. SWITCHES

Simple Switches

A simple and common example of a switching device is a manually operated mechanical *switch*, e.g., a light switch or a push button. Associated with a simple switch are two terminals, t_1 and t_2, which are connected (short-circuited) when the switch is operated, and disconnected (open-circuited) when it is unoperated. A schematic diagram of a simple switch is shown in Fig. 1-1(a). At any time the switch is either operated or unoperated; consequently, its behavior can be described by a *binary*, i.e., two-valued, variable s.

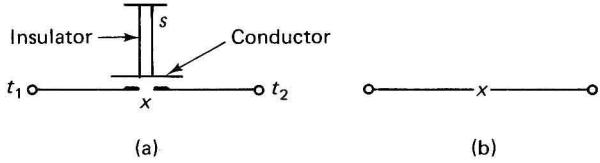

Fig. 1-1 (a) Schematic diagram of a simple switch; (b) symbolic diagram.

We use the convention that $s = 0$ when the switch is unoperated and $s = 1$ when it is operated. The choice of symbols 0 and 1 is arbitrary and the symbols have no numerical significance. Any other pair of distinct symbols would be equally acceptable. For the terminals t_1 and t_2 we introduce a binary variable x such that $x = 0$ if the path between t_1 and t_2 is open, and $x = 1$ if it is closed. Thus the performance of the switch of Fig. 1-1(a) can be specified by the statement $x = 1 \equiv s = 1$, or simply by $x = s$. To simplify the representation of switches we use the symbolic diagram of Fig. 1-1(b) rather than the schematic drawing.

A slightly more complicated switch is shown in Fig. 1-2, where two pairs

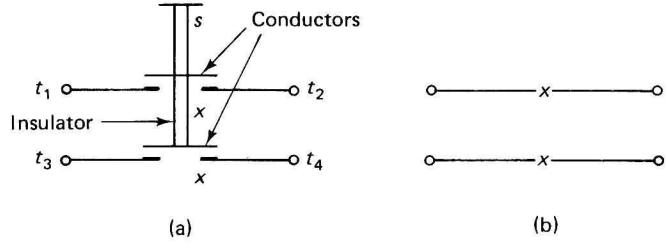

Fig. 1-2 Illustrating multiple contacts.