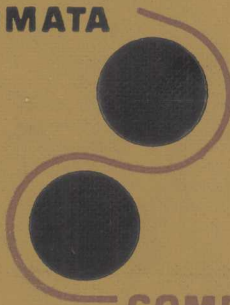


**SWITCHING AND FINITE AUTOMATA  
THEORY**



**COMPUTER  
SCIENCE  
SERIES**

6  
8093575

# SWITCHING AND FINITE AUTOMATA THEORY

Second Edition

ZVI KOHAVI

Departments of Electrical Engineering  
and Computer Science

TECHNION-Israel Institute of Technology



McGraw-Hill Book Company

New York St. Louis San Francisco Auckland Bogotá Düsseldorf  
Johannesburg London Madrid Mexico Montreal New Delhi Panama  
Paris São Paulo Singapore Sydney Tokyo Toronto

337701



## SWITCHING AND FINITE AUTOMATA THEORY

Copyright © 1978, 1970 by McGraw-Hill, Inc.  
All rights reserved.

Printed in the United States of America.

No part of this publication may be reproduced,  
stored in a retrieval system, or transmitted,  
in any form or by any means, electronic,  
mechanical, photocopying, recording, or otherwise,  
without the prior written permission of the publisher.

567890 DODO 83210

This book was set in Modern 8A by Bi-Comp, Incorporated.  
The editors were Julianne V. Brown and Madelaine Eichberg;  
the production supervisor was Charles Hess.  
New drawings were done by Fine Line Illustrators, Inc.  
R. R. Donnelley & Sons Company was printer and binder.

### Library of Congress Cataloging in Publication Data

Kohavi, Zvi.

Switching and finite automata theory.

(McGraw-Hill computer science series)

Includes bibliographical references and index.

1. Sequential machine theory. 2. Switching theory.

I. Title.

QA267.5.S4K64 1978

629.8'9

77-13703

ISBN 0-07-035310-7

**SWITCHING AND FINITE  
AUTOMATA THEORY**

**McGRAW-HILL COMPUTER SCIENCE SERIES**

EDWARD A. FEIGENBAUM: *Stanford University*

RICHARD W. HAMMING: *Naval Post Graduate School*

ALLEN: *Anatomy of LISP*

BELL AND NEWELL: *Computer Structures: Readings and Examples*

DONOVAN: *Systems Programming*

GEAR: *Computer Organization and Programming*

GIVONE: *Introduction to Switching Circuit Theory*

GOODMAN AND HEDETNIEMI: *Introduction to the Design  
and Analysis of Algorithms*

HAMACHER, VRANESIC, AND ZAKY: *Computer Organization*

HAMMING: *Introduction to Applied Numerical Analysis*

HAYES: *Computer Architecture and Organization*

HELLERMAN: *Digital Computer System Principles*

HELLERMAN AND CONROY: *Computer System Performance*

KAIN: *Automata Theory: Machines and Languages*

KATZAN: *Microprogramming Primer*

KOHAVI: *Switching and Finite Automata Theory*

LIU: *Elements of Discrete Mathematics*

LIU: *Introduction to Combinatorial Mathematics*

MADNICK AND DONOVAN: *Operating Systems*

MANNA: *Mathematical Theory of Computation*

NEWMAN AND SPROULL: *Principles of Interactive Computer Graphics*

NILSSON: *Problem-solving Methods in Artificial Intelligence*

ROSEN: *Programming Systems and Languages*

SALTON: *Automatic Information Organization and Retrieval*

STONE: *Introduction to Computer Organization and Data Structures*

STONE AND SIEWIOREK: *Introduction to Computer Organization  
and Data Structures: PDP-11 Edition*

TONGE AND FELDMAN: *Computing: An Introduction to Procedures and  
Procedure-Followers*

TREMBLAY AND MANOHAR: *Discrete Mathematical Structures  
with Applications to Computer Science*

TREMBLAY AND SORENSON: *An Introduction to Data Structures  
with Applications*

TUCKER: *Programming Languages*

WATSON: *Timesharing System Design Concepts*

WEGNER: *Programming Languages, Information Structures,  
and Machine Organization*

WIEDERHOLD: *Database Design*

WINSTON: *The Psychology of Computer Vision*

# Preface

The subject of switching and finite automata theory needs no introduction. It has become a part of every computer science and electrical engineering curriculum, and rightly so. It provides techniques useful in a wide variety of applications and helps develop *a way of thinking* that leads to understanding of the structure, behavior, and limitations and capabilities of logical machines. In this book I have tried to cover the whole subject, starting with introductory material and leading to the more advanced topics, assuming a minimal technical background on the part of the reader. I did not attempt to provide detailed techniques for the design of specific circuits, but rather to formulate methods and to develop algorithms that can be applied to a broad class of problems. For once such general principles are understood, the relevance of specific procedures and their applicability to given problems are a matter of engineering decisions.

I have endeavored to provide a logical and rigorous presentation with a minimum of formalism. Accordingly, theorems are proved and algorithms are carefully developed, but only after an intuitive understanding of the procedures involved has been achieved by means of illustrative examples. Throughout I have assumed that the reader of this book is a computer scientist, a logical designer, or a communication or control engineer. Applications and examples are drawn accordingly from these fields. Most of the material is not new, although many subjects appear for the first time in an introductory text. A listing of the main sources upon which I have drawn, as well as some historical notes, is provided at the end of each chapter. These references are just the basic ones, and should not be considered a comprehensive bibliography.

The book is divided into three parts. The first part, which consists of Chapters 1 and 2, provides some introductory background. The second part is devoted to combinational logic, and the third part is concerned with finite automata. The book is organized so that many of its chapters cover specific topics and they are not prerequisite for subsequent chapters.

In this category are Chapters 6, 7, 8, 11, 12, 13, 14, 15, 16, and their selection in a course outline depends on the preferences of the instructor. Sections marked with a star (★) may be omitted without loss of continuity.

The book is self-contained as a text. At present much of this material is customarily taught in graduate courses, but from my experience at the Polytechnic Institute of New York, the Massachusetts Institute of Technology, and the Technion most of it can be taught at the junior or senior level in computer science or electrical engineering departments. The book is intended as a text for a two-semester sequence. The first semester can be devoted to "classical" switching theory [Chapters 1, 3 (Sections 3-1 to 3-4), and 4 through 11], and the second semester devoted primarily to finite automata theory [Chapters 2, and 12 through 16]. Other organizations of the material into one-semester or two-semester courses are possible, keeping in mind the following prerequisites: Chapters 3 (excluding Section 3-5), 4, and 5 are prerequisite for the entire book. Chapters 9 and 10 (excluding Section 10-4) are prerequisite for Chapters 12 through 16.

In revising the book some chapters have undergone a major revision, while others required only minor changes. Chapters 5 and 9 have been updated to reflect the general use of integrated circuits in logical design. Chapters 8 and 13 have been revised and expanded to reflect the importance of testing circuits and of designing more reliable circuits. In an era where computers are used to execute algorithms and perform experiments, it is of utmost importance to establish that the computations will indeed terminate and to provide bounds for their length. The development of bounds on the length of various experiments has therefore been further emphasized by providing new proofs in appendixes and problems in Chapters 10, 13, and 14. The problem sets, which were originally quite extensive, have been further expanded. They range from simple numerical examples to natural extensions of the ideas presented in the text.

The first edition of this book has been used in many universities and departments and I received numerous helpful comments. I am grateful for all of them. My gratitude is due to M. Yoeli of the Technion and E. J. Smith of the Polytechnic Institute of New York, who stimulated my first interest in the subject of switching and automata theory. I also wish to thank I. Kohavi, I. Koren, S. Patil, R. Riesenfeld, and C. L. Seitz for their helpful comments and many valuable suggestions. I am indebted to the electrical engineering department at M.I.T. which provided a stimulating atmosphere for writing the first edition, and to the Technion and the computer science department at the University of Utah for making the revised edition possible.

Special gratitude is due to C. L. Liu of the University of Illinois for his thorough review of the entire manuscript and his invaluable discerning technical criticism, which significantly improved this book. I also thank Karen Evans for her excellent typing of the manuscript and for drawing the figures. Last but not least, I wish to thank my wife, Sima, for her help and understanding.

Zvi Kohavi



6  
8003575  
5

# Contents



## Preface

xi

## PART ONE PRELIMINARIES

<b>Chapter 1</b>	<b>Number Systems and Codes</b>	<b>3</b>
1-1	Number Systems	3
1-2	Binary Codes	10
1-3	Error Detection and Correction	14
	Notes and References	20
	Problems	21

<b>Chapter 2</b>	<b>Sets, Relations, and Lattices</b>	<b>24</b>
2-1	Sets	24
2-2	Relations	26
2-3	Partially Ordered Sets	30
2-4	Lattices	33
	Notes and References	36
	Problems	36

## PART TWO COMBINATIONAL LOGIC



E8053575

<b>Chapter 3</b>	<b>Switching Algebra and Its Applications</b>	<b>41</b>
3-1	Switching Algebra	42
3-2	Switching Functions	48
3-3	Isomorphic Systems	58
3-4	Electronic-Gate Networks	63

★ 3-5	Boolean Algebras	64
	Notes and References	66
	Problems	67
Chapter 4 Minimization of Switching Functions		74
4-1	Introduction	74
4-2	The Map Method	76
4-3	Minimal Functions and Their Properties	86
4-4	The Tabulation Procedure for the Determination of Prime Implicants	90
4-5	The Prime Implicant Chart	95
4-6	Map-entered Variables	103
	Notes and References	105
	Problems	106
Chapter 5 Logical Design		113
5-1	Design with Basic Logical Gates	114
5-2	Logic Design with Integrated Circuits	119
5-3	NAND and NOR Circuits	134
5-4	Design of High-Speed Adders	138
5-5	Relay Contacts	144
5-6	Analysis and Synthesis of Contact Networks	145
	Notes and References	153
	Problems	154
Chapter 6 Functional Decomposition and Symmetric Functions		162
6-1	Functional Decomposition	162
6-2	Symmetric Networks	173
6-3	Identification of Symmetric Functions	180
	Notes and References	184
	Problems	185
Chapter 7 Threshold Logic		189
7-1	Introductory Concepts	190
7-2	Synthesis of Threshold Networks	197
	Notes and References	209
	Problems	209

★ Sections marked with a star may be omitted without loss of continuity.

<b>Chapter 8</b>	<b>Reliable Design and Fault Diagnosis</b>	<b>214</b>
8-1	Hazards	214
8-2	Fault Detection in Combinational Circuits	218
8-3	Fault-Location Experiments	223
8-4	Boolean Differences	228
8-5	Fault Detection by Path Sensitizing	234
8-6	Detection of Multiple Faults	238
8-7	Failure-Tolerant Design	250
8-8	Quadded Logic	258
	Notes and References	263
	Problems	265

### PART THREE FINITE-STATE MACHINES

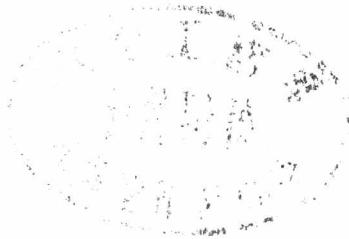
<b>Chapter 9</b>	<b>Introduction to Synchronous Sequential Circuits and Iterative Networks</b>	<b>275</b>
9-1	Sequential Circuits—Introductory Example	276
9-2	The Finite-State Model—Basic Definitions	280
9-3	Memory Elements and Their Excitation Functions	283
9-4	Synthesis of Synchronous Sequential Circuits	291
9-5	An Example of a Computing Machine	305
9-6	Iterative Networks	309
	Notes and References	315
	Problems	315
<b>Chapter 10</b>	<b>Capabilities, Minimization, and Transformation of Sequential Machines</b>	<b>322</b>
10-1	The Finite-State Model—Further Definitions	322
10-2	Capabilities and Limitations of Finite-State Machines	325
10-3	State Equivalence and Machine Minimization	327
10-4	Simplification of Incompletely Specified Machines	333
	Notes and References	348
	Problems	348
<b>Chapter 11</b>	<b>Asynchronous Sequential Circuits</b>	<b>356</b>
11-1	Fundamental-Mode Circuits	356
11-2	Synthesis	358

11-3	State Assignment in Asynchronous Sequential Circuits	368
11-4	Pulse-Mode Circuits	373
	Notes and References	375
	Problems	376
<b>Chapter 12 Structure of Sequential Machines</b>		<b>385</b>
12-1	Introductory Example	385
12-2	State Assignments Using Partitions	388
12-3	The Lattice of Closed Partitions	394
12-4	Reduction of the Output Dependency	397
12-5	Input Independence and Autonomous Clocks	400
12-6	Covers and Generation of Closed Partitions by State Splitting	403
12-7	Information Flow in Sequential Machines	411
12-8	Decomposition	420
★ 12-9	Synthesis of Multiple Machines	430
	Notes and References	435
	Problems	437
<b>Chapter 13 State-Identification and Fault-Detection Experiments</b>		<b>449</b>
13-1	Experiments	450
13-2	Homing Experiments	454
13-3	Distinguishing Experiments	458
13-4	Machine Identification	463
13-5	Fault-Detection Experiments	464
★ 13-6	Design of Diagnosable Machines	470
★ 13-7	Second Algorithm for the Design of Fault-Detection Experiments	476
★ 13-8	Fault-Detection Experiments for Machines Which Have no Distinguishing Sequences	479
Appendix 13-1	Bounds on the Length of Synchronizing Sequences	491
Appendix 13-2	Bounds on the Length of Distinguishing Sequences	495
	Notes and References	495
	Problems	496
<b>Chapter 14 Memory, Definiteness, and Information Losslessness of Finite Automata</b>		<b>507</b>
14-1	Memory Span with Respect to Input-Output Sequences (Finite-Memory Machines)	508

14-2	Memory Span with Respect to Input Sequences (Definite Machines)	513
14-3	Memory Span with Respect to Output Sequences	518
14-4	Information Lossless Machines	522
★14-5	Synchronizable and Uniquely Decipherable Codes	536
Appendix 14-1	The Least Upper Bound for Information Losslessness of Finite Order	543
	Notes and References	546
	Problems	547
Chapter 15	Linear Sequential Machines	557
15-1	Introduction	557
15-2	Inert Linear Machines	560
15-3	Inert Linear Machines and Rational Transfer Functions	567
15-4	The General Model	573
15-5	Reduction of Linear Machines	578
15-6	Identification of Linear Machines	587
15-7	Application of Linear Machines to Error Correction	593
Appendix 15-1	Basic Properties of Finite Fields	596
Appendix 15-2	The Euclidean Algorithm	598
	Notes and References	599
	Problems	600
Chapter 16	Finite-State Recognizers	609
16-1	Deterministic Recognizers	610
16-2	Transition Graphs	611
16-3	Converting Nondeterministic into Deterministic Graphs	614
16-4	Regular Expressions	617
16-5	Transition Graphs Recognizing Regular Sets	623
16-6	Regular Sets Corresponding to Transition Graphs	631
★16-7	Two-Way Recognizers	638
	Notes and References	645
	Problems	646
Index		653

## **SWITCHING AND FINITE AUTOMATA THEORY**

AT 7/7/67



Part one  
Preliminaries

# Number Systems and Codes



This chapter deals with the representation of numerical data, with emphasis on those representations which use only two symbols, 0 and 1. Described are special methods of representing numerical data, affording protection against various transmission errors and component failures.

## 1-1 NUMBER SYSTEMS

Convenient as the decimal number system generally is, its usefulness in machine computation is limited because of the nature of practical electronic devices. In most present digital machines the numbers are represented, and the arithmetic operations performed, in a different number system, called the binary number system. This section is concerned with the representation of numbers in various systems and with methods of conversion from one system to another.

### Number representation

An ordinary decimal number actually represents a polynomial in powers of 10. For example, the number 123.45 represents the polynomial

$$123.45 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 + 4 \cdot 10^{-1} + 5 \cdot 10^{-2}$$



This method of representing decimal numbers is known as the *decimal number system*, and the number 10 is referred to as the *base* (or *radix*) of the system. In a system whose base is  $b$ , a positive number  $N$  represents the polynomial

$$\begin{aligned} N &= a_{q-1}b^{q-1} + \cdots + a_0b^0 + \cdots + a_{-p}b^{-p} \\ &= \sum_{i=-p}^{q-1} a_i b^i \end{aligned}$$

where the base  $b$  is an integer greater than 1, and the  $a$ 's are integers in the range  $0 \leq a_i \leq b - 1$ . The sequence of digits  $a_{q-1}a_{q-2} \cdots a_0$  constitutes the *integral part* of  $N$ , while the sequence  $a_{-1}a_{-2} \cdots a_{-p}$  constitutes the *fractional part* of  $N$ . Thus  $p$  and  $q$  designate the number of digits in the fractional and integral parts, respectively. The integral and fractional parts are usually separated by a *radix point*. The digit  $a_{-p}$  is referred to as the *least significant digit*, while  $a_{q-1}$  is called the *most significant digit*.

When the base  $b$  equals 2, the number representation is referred to as the *binary number system*. For example, the binary number 1101.01 represents the polynomial

$$1101.01 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

that is,

$$1101.01 = \sum_{i=-2}^3 a_i 2^i$$

where  $a_{-2} = a_0 = a_2 = a_3 = 1$  and  $a_{-1} = a_1 = 0$ .

A number  $N$  in base  $b$  is usually denoted  $(N)_b$ . Whenever the base is not specified, base 10 is implicit. Table 1-1 shows the representations of integers 0 through 15 in several number systems.

The *complement* of a digit  $a$ , denoted  $a'$ , in base  $b$  is defined as

$$a' = (b - 1) - a$$

That is, the complement  $a'$  is the difference between the largest digit in base  $b$  and the digit  $a$ . In the binary number system, since  $b = 2$ ,  $0' = 1$  and  $1' = 0$ . In the decimal number system the largest digit is 9. Thus, for example, the complement† of 3 is  $9 - 3 = 6$ .

### Conversion of bases

Suppose that some number  $N$ , which we wish to express in base  $b_2$ , is presently expressed in base  $b_1$ . In converting a number from base  $b_1$  to

† In the decimal system the complement is also referred to as the *9's complement*. In the binary system it is also known as the *1's complement*.