

DAVID C. RINE, Editor

COMPUTER SCIENCE AND MULTIPLE- VALUED LOGIC

THEORY AND APPLICATIONS

NORTH-HOLLAND

TP301
R579

7960629

5

computer science and multiple-valued logic theory and applications

edited by

david c. rine



1977



E7950629

north-holland publishing company—amsterdam · new york · oxford

TP
C

© North-Holland Publishing Company - 1977

All rights reserved. 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 prior permission of the publisher.

Library of Congress Catalog Card Number: 75-40064

North-Holland ISBN: 0 7204 0406 1

Published by:

North-Holland Publishing Company - Amsterdam, New York, Oxford

Sole distributors for the U.S.A. and Canada:

Elsevier/North-Holland Inc.

52 Vanderbilt Avenue

New York, NY 10017

Library of Congress Cataloging in Publication Data

Main entry under title:

Computer science and multiple valued logic.

1. Switching theory. 2. Logic circuits. 3. Threshold logic. 4. Many-valued logic. I. Rine, David C.

QA268.5.C65 511'.3 75-40064

ISBN 0-444-11056-6 (American Elsevier)

Printed in England

**computer science and
multiple-valued logic**
theory and applications

list of contributors

- George ABRAHAM, Naval Research Laboratory, Washington, DC, U.S.A.
C. Michael ALLEN, Dept. Electrical Engineering, University of North Carolina at Charlotte, Charlotte, NC 28213, U.S.A.
Robert C. BRADDOCK, ITT-Gilfillan Incorporated, 7821 Orion Avenue, Van Nuys, CA 91409, U.S.A.
Melvin A. BREUER, Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90007, U.S.A.
Peter T. CHEUNG, Packard Instrument Co., Downers Grove, IL, U.S.A.
Edgar DuCASSE, Dept. of Computer and Information Science, Brooklyn College, C.U.N.Y., Brooklyn, New York 11236, U.S.A.
George EPSTEIN, Indiana University, Bloomington, IN 47401, U.S.A.
Gideon FRIEDER, Scientific Center, IBM Israel Ltd., Technion City, Haifa, Israel.
Donald D. GIVONE, Dept. of Electrical Engineering, Parker Engineering Building, State University of New York at Buffalo, Buffalo, NY 14214, U.S.A.
V. C. HAMACHER, Depts. of Electrical Engineering & Computer Science, University of Toronto, Canada.
A. HORN, University of Southern California, Los Angeles, CA 90007, U.S.A.
Samuel C. LEE, Dept. Electrical Engineering, University of Oklahoma, Norman, OK 73069, U.S.A.
Marvin E. LIEBLER, Westinghouse, Buffalo, NY, U.S.A.
Ryszard S. MICHALSKI, Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.
Claudio MORAGA, Dept. of Informatics, University of Dortmund, 46 Dortmund 50, F.R.G.
David C. RINE, Computer Science, University of Texas, San Antonio, TX, U.S.A.
Robert P. ROESSER, Dept. of Electrical Engineering, Wayne State University, Detroit, MI 48202, U.S.A.
Ivo G. ROSENBERG, University of Montreal, C.P.6128 Montreal 101, Quebec, Canada.
K. C. SMITH, Depts. of Electrical Engineering & Computer Science, University of Toronto, Canada.
William R. SMITH, Dept. Electrical Engineering, University of Bridgeport, CT 06602, U.S.A.
Stephen Y. H. SU, Dept. of Electrical Engineering, Utah State University, Logan, UT 84322, U.S.A.
Stanislaw J. SURMA, Dept. of Logic, Jagiellonian University of Cracow, ul. Grodska 52, 31-044 Krakow, Poland.

T. TRACZYK, Koszkoma 75.m.27, 00-662 Warszawa, Poland.

Z. G. VRANESIC, Depts. of Electrical Engineering & Computer Science, University of Toronto, Canada.

Anthony S. WOJCIK, Dept. of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, U.S.A.

Hideki YAMANAKA, ITT-Gilfillan Incorporated, 7821 Orion Avenue, Van Nuys, CA 91409, U.S.A.

preface

This preface briefly acquaints the reader to the book and its preparation, whereas the introductory chapter introduces the reader to the area of multiple-valued logic and how and why it has become related to computer science.

This book is aimed at those computer engineers, computer scientists, applied mathematicians and physicists who are not experts in the area of multiple-valued logic as the discipline pertains to computer engineering and computer science. Thus its primary goal is to convince these people that multiple-valued logic is indeed both intellectually challenging and practically applicable.

It was decided in conversations with a number of engineers, computer scientists and potentially interested people including applied mathematicians that several outstanding scientists, luminaries capable of explaining our basic concepts in an interesting way, should be asked to write an updated 'handbook' of ideas contained in the best papers. With the exception of a handful of good classical works by Epstein this expectation has been carried out.

Many new developments have recently taken place in computer science and multiple-valued logic. For example, fault analysis and software considerations are new and extremely important fields. In fact, there is a good chance that the impact of multiple-valued software considerations will be at least as important as hardware considerations on the use and design of fourth and fifth generation computing and data processing systems.

The organization of this documentary is top-down in its design in that the book is divided into five parts.

I Algebraic theory.

II Logic design and switching theory.

III Threshold logic design.

IV Physical components and implementations.

V Applications.

Each part has been broken down into several monographs or chapters, each chapter having been designed by one or two scientists. However the flow and sense from chapter to chapter is continuous. The flow between parts has been made continuous by means of introductory remarks and conclusions for each section.

The Introduction (Chapter 1) provides the new reader with interesting foundations, history and motivation for considering multiple-valued logic design and applications from both intellectual and practical points of view. No particular advanced technical background is assumed of the reader; likewise, the first portion of each monograph is written in such a way that important concepts or reasons for considering a given area can be grasped by each new reader. Since the

book relates to computer science and engineering, many interesting considerations from ancient history have not been mentioned due to lack of space. However, a conspectus on the history can be found in Rescher[41]. A short but precise historical summary of multiple-valued logic as it is related to computer science can be found in a paper by Epstein, Frieder, and Rine[14]. Papers which could date the beginning of multiple-valued logic and modern digital computer systems are those of Groth[20] and Metze[31].

Part I on algebraic theory gives the applied mathematical foundations for much of multiple-valued logic as related to computer science. The reader who is either partially familiar with these interesting relations or who is more immediately interested in only logic design and switching theory may wish to turn directly to Part II. Modern axioms and properties of Post algebras are nicely summarized by Traczyk[54], a Polish scientist; while Surma[53] presents a general algorithm for axiomatizing every finite logic. Relationships between Post and Boolean algebras have been studied by Wojcik and Metze[59,60]. Completeness properties of multiple-valued logic and switching theory have been documented by Rosenberg[45] and Patt[35]. Abstractions and extensions of Post algebras have recently been introduced by Epstein[11, 12, 14], one interesting concept being that of a P-algebra; Epstein has, also, given an important equational axiomatization for the disjoint system of Post algebras[13]. Interestingly enough, Nutter, Swartwout, and Rine[37] have shown that many multiple-valued logic algebras are isomorphically equivalent to Post algebras.

Part II of this book covers logic design and switching theory. This section gives important models for computer subsystems and evaluates multiple-valued logic design criteria directly related to cost. Initially, applications of multiple-valued logic systems to circuits were studied by Metze[31] and Yoeli and Rosenfeld[61]. Since various algorithms for minimizing Boolean logic systems, for example the Quine-McCluskey technique, have been of interest, it must be noted that much research has also been directed toward the "minimizing" of multiple-valued logic systems. While this general problem is still open and is far more difficult for the multiple-valued logic cases and presents itself as a "can-of-worms", computer minimization techniques have seen partial success through the ongoing work of Allen and Givone[3], Su and Cheung[52], W. R. Smith[50], and Ostapko, Cain, and Hong[38]. Hazards and cost analyses of hazard-free logic systems have always been an important topic of investigation in computer engineering. DuCasse and Metze[10] and others[24] are studying this problem. Also, fault detection using multiple-valued test machines has been studied by Sheppard and Vranesic[40], while a multilevel balanced code with redundant digits was investigated by Asabe and Tezuka[4]. Treatment of error signals was reported by Breuer and Epstein[7]. Finally, multiple-valued logic symmetric functions and their properties were well-noted by S. C. and E. T. Lee[28] and also by Cheung and Su[8].

Threshold Logic Design, covered in Part III, is an extremely important area of

computer engineering and logic design in its own right. Multiple-valued variable-threshold logic is being investigated by Aibara, Takamatsu, and Murakami, while applications of multiple-valued threshold logic to digital computer arithmetic were discovered by the Finnish computer engineers, Koukkunen and Ojala[26]. On the other hand, Druzeta and Sedra[9] have made good progress by using multiple-threshold circuits in the design of multi-state storage elements. Finally, it must be mentioned that Moraga has achieved excellent results in the study of non-linear[33], minimal[37], and periodic[34] ternary threshold logic systems. Another good summary paper is that of Vranesic and Hamacher on threshold logic in fast ternary multipliers[50]. Along the same lines is the work on multifunction threshold gates by Hampel[23].

It goes without saying from a commercial-industrial point of view that Part IV, physical components and implementations, is very important for computer engineering. Historically, despite relatively little effort being put into the design of electronic circuits for multiple-valued logic, there has been a very encouraging evolutionary trend towards many more practical realizations. In the past several years implementations have suffered from insufficient utilization of integrated circuit techniques. In the past, for example, stray capacitance in complex discrete circuits has led to slow operating characteristics. However, additional components and the miniaturization possible with integrated circuit technology is now beginning to lead to a much improved speed performance. Once the speed problems are overcome, the natural and very valuable advantages of multiple-valued logic techniques in the reduction of wiring complexity, a big cost factor in building present day computers, will assert themselves. The computer engineering group at the University of Toronto has made elegant strides in constructing and implementing multiple-valued logic physical components; I mention the work of Vranesic and Smith[56,57] and Sebastian and Vranesic[47]. One must also mention the independent work on multiple-valued integrated circuits by Abraham[1] at the Naval Research Laboratory. A new concept for ternary logic elements has been reported by Etienne and Israel[16], while Mouftah and Jordan[35] have reported success on integrated circuits for ternary logic. Shiva and Nagle[49] at Auburn University have advanced technology on multiple-valued memory elements, while Strasilla[57] in Switzerland has made excellent headway on a multiple-valued logic memory system using capacitor storage. Also in the area of physical components Irving and Nagle[25] have invented an approach to multiple-valued sequential logic. Historically, Braddock, Epstein, and Yamanaka[6] hold an original patent on a multiple-valued logic design and application in binary computers. The work of Metze[31] also deserves mention in this section.

Part V, applications, is one of the most important because it covers the ways in which people use multiple-valued logic systems in problem solving. This section describes both hardware and software applications in the very new area of multiple-valued logic and programming. This area will see a major expansion

within the next few years. Microprogramming has made it possible sensibly to study the emulation of multiple-valued logic computer systems; it is often better to build an emulator than an actual prototype of the system. It has been said that "emulation is the bridge between hardware and software". Frieder, Fong, Chao, and Luk[18,19] have used emulation to study a balanced ternary computer, particularly the arithmetic unit. On the other hand, Halpern and Yoeli[21] and Vranesic and Hamacher[22,55] consider the actual hardware prototype of the arithmetic unit in their studies. Another good German reference is [5]. Of importance to testing in computer engineering, Rowe[46] has researched the generation by multiple-valued logic of pseudorandom noise, while Lam and Vranesic[27] reported the multiple-valued logic generation of pseudorandom numbers, potentially better than binary generation. Leibler and Roesser[29] describe multiple-real-valued Walsh functions, while McDonald and Singh[30] have extensively researched the extensions of the Weiner-Smith algorithm for multiple-valued logic synchronous sequential circuit design. Asabe and Tezuka[4] reported on the multilevel balanced code with redundant digits. Returning to software applications, Rine[42,43] has mentioned applications of multiple-valued logic to programming languages and also to statistical decision theory[44]. Of great interest to computer science, diagnostic pattern recognition and multiple-valued logic systems has been the variable-valued logic systems VL1 Project under the direction of R. S. Michalski[32]. Also of interest has been the recent work of Rasiowa[40] on a logical structure of mix-valued programs.

References

- [1] G. Abraham, Variable radix multistable integrated circuits: Impact of electron physics on multiple-valued logic design, 1974 *Intern. Symp. on Multiple-Valued Logic, and Computer*, September 1974.
- [2] T. Aibara, Y. Takamatsu, and K. Murakami, Multiple-valued variable-threshold logic and its application to the realization of the set of boolean functions, *Proc. 1973 Intern. Symp. on Multiple-Valued Logic*, Toronto, Canada.
- [3] C. M. Allen and D. D. Givone, A minimization technique for multiple-valued logic systems, *IEEE Trans. Computers*, C-17 (1968) 182-184.
- [4] T. Asabe and Y. Tezuka, Multilevel balanced code with redundant digits, *Proc. 1972 Symp. Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [5] Arithmetisch Operationen Mit Binär Codierten Ternarzahlen, *AGEN*, 11 (1970) 55.
- [6] R. Braddock, G. Epstein, and H. Yamanaka, Multiple-valued logic design and applications in binary computers, *Proc. 1971 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [7] M. A. Breuer and G. Epstein, The smallest many-valued logic for treatment of complemented and uncomplemented error signals, *Proc. 1973 Intern. Symp. on Multiple-Valued Logic*, Toronto, Canada.
- [8] P. Cheung and S. Su, Algebraic properties of multi-valued symmetric switching functions, *Proc. 1971 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [9] A. Druzeta and A. S. Sedra, Multi-threshold circuits in the design of multi-state storage elements, *Proc. 1973 Intern. Symp. on Multiple-Valued Logic*, Toronto, Canada.
- [10] E. G. DuCasse and G. A. Metze, Hazard-free realizations of boolean functions using post functions (Also, A cost analysis of hazard-free networks synthesized from monotone jump

functions), *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W.Va.

- [11] G. Epstein, On multiple-valued signal processing with limiting, *Proc. 1971 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [12] G. Epstein, P-Algebras, an abstraction from Post algebras, Tech. Rep., U.C.L.A. and The Indiana University, Bloomington, (1973).
- [13] G. Epstein, An equational axiomatization for the disjoint system of Post algebras, *IEEE Trans. Computers* C-22 (1973) 422-423.
- [14] G. Epstein, G. Frieder and D. Rine, The development of multiple-valued logic as related to computer science: A historical summary, *Computer*, September 1974. (And, a brief in *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.).
- [15] G. Epstein and A. Horn, Chain based lattices, *Pac. J. Math* (1975).
- [16] D. Etienne and M. Israel, A new concept for ternary logic elements, *Proc. 1974 Intern. Symp. Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [17] Final report on design of multiple-valued logic systems, The Digital Systems Labs., SUNY at Buffalo, N.Y. (June, 1970).
- [18] G. Frieder, A. Fong and C. Y. Chao, A balanced ternary computer, *Proc. 1973 Intern. Symp. on Multiple-Valued Logic*, Toronto, Canada.
- [19] G. Frieder and C. Luk, Algorithms for binary coded balanced and ordinary ternary operations, SUNY Computer Science Report, Buffalo, N.Y. (1974).
- [20] H. R. Groth, Signed ternary arithmetic, Digital Computer Lab., MIT, Cambridge, Mass., Memorandum M-1496 (May 1954).
- [21] I. Halpern and M. Yoeli, Ternary Arithmetic Unit, *Proc. IEEE*, 115 (1968) 1385-1388.
- [22] V. C. Hamacher and Z. Vranesic, Multivalued versus binary high speed multipliers, *Proc. 1971 Symp. on Theory and Application of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [23] D. Hampel, Multifunction Threshold Gates, *IEEE Trans. Computers*, C-22 (1973) 197-203.
- [24] Hazard Detection in Combinational and Sequential Switching Circuits, *IBM J. Res. Develop.*, 9 (1965) 90-99.
- [25] T. A. Irving and H. T. Nagle, An approach to multi-valued sequential logic, *Proc. 1973 Intern. Symp. on Multiple-Valued Logic*, Toronto, Canada.
- [26] H. Koukkunen and L. Ojala, Some applications of many-valued threshold logic in digital arithmetic, *Proc. 1972 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [27] C. L. Lam and Z. G. Vranesic, Multiple-valued pseudorandom number generators, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. VA.
- [28] S. C. Lee and E. T. Lee, On Multivalued Symmetric Functions, *IEEE Trans. Computers*, March 1972.
- [29] M. Leibler and R. Roesser, Multiple-real-valued Walsh functions, *Proc. 1971 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [30] J. F. McDonald and I. Singh, Extensions of the Weiner-Smith algorithm for m -ary synchronous sequential circuit design, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [31] G. Metze, An Application of Multivalued Logic Systems to Circuits, *Proc. Symp. on Circuit Analysis*, University of Illinois, (1955) pp. 11-1-11-14.
- [32] R. S. Michalski, Variable-valued logic systems-VL1, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [33] C. Moraga, Non-linear ternary threshold logic, *Proc. 1972 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [34] C. Moraga and J. Gutierrez, Multithreshold periodic ternary threshold logic, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [35] H. T. Mouftah and I. B. Jordan, Integrated circuits for ternary logic, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.

- [36] J. Nazarala and C. Moraga, Minimal realization of ternary threshold functions, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [37] R. Nutter, R. Swartwout and D. Rine, Equivalence and transformations for Post multivalued algebras, *IEEE Trans. Computers* C-23 (1974) 294-300.
- [38] D. L. Ostapko, R. G. Cain and S. J. Hong, A practical approach to two-level minimization of multivalued logic, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [39] Y. N. Pratt, Toward a characterization of logically complete switching functions in K-valued logic, *Proc. 1972 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [40] H. Rasiowa, On a logical structure of mix-valued programs and the ω^+ -valued algorithmic logic, *Bull. Acad. Polon. Sci. Ser. Sci. Math. Astronom. Phys.* 21 (1973).
- [41] N. Rescher, *A Historical Conspectus of Many-Valued Logics*, (McGraw-Hill, N.Y., 1969).
- [42] D. C. Rine, A proposed multi-valued extension to ALGOL 68, *Kybernetes* 2 (1973) 107-111.
- [43] D. C. Rine, Conditional and Post algebra expressions, *Discrete Math.* 10 (1974) 309-324.
- [44] D. C. Rine, Basic concepts of multiple-discrete valued logic and decision theory, *Information and Control* 22 (1973) 320-352.
- [45] I. G. Rosenberg, Completeness, closed classes and relations in multiple-valued logics, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [46] I. Rowe, The generation of Gaussian-distributed pseudorandom noise sequences from multiple-valued logic, *Proc. 1972 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [47] P. Sebastian and Z. G. Vranesic, Ternary logic in arithmetic units, *Proc. 1972 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [48] D. A. Sheppard and Z. G. Vranesic, Fault detection of binary sequential machines using R-valued test machines, *IEEE Trans. Computers* C-23 (1974) 352-358.
- [49] S. G. Shiva and H. T. Nagle, On multiple-valued memory elements, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [50] W. R. Smith, Minimization of multivalued functions, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [51] U. J. Strasilla, A multiple-valued logic memory system using capacitor storage, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [52] S. Y. H. Su and P. T. Cheung, Computer minimization of multivalued switching functions, *IEEE Trans. Computers* C-21 (1972) 995-1003.
- [53] S. J. Surma, An algorithm for axiomatizing every finite logic, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va. (and in *Reports on Mathematical Logic* no. 3 (1974) p. 57.).
- [54] T. Traczyk, Axioms and some properties of Post algebras, *Colloq. Math.* 10 (1963) 193-209.
- [55] Z. G. Vranesic and V. C. Hamacher, Ternary logic in Parallel multipliers, *Comput. J.* 15 (1972) 254-258.
- [56] Z. G. Vranesic and K. C. Smith, An electronic implementation of multiple-valued logic, *Proc. 1974 Intern. Symp. on Multiple-Valued Logic*, West Virginia University, Morgantown, W. Va.
- [57] Z. G. Vranesic and K. C. Smith, Engineering aspects of multi-valued logic systems, *Computer*, September 1974.
- [58] Z. G. Vranesic and V. C. Hamacher, Threshold logic in fast ternary multipliers, *Proc. 1975 Intern. Symp. on Multiple-Valued Logic*, University of Indiana, Bloomington, Ind.
- [59] A. Wojcik, The minimization of higher-order Boolean functions, *Proc. 1972 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, N.Y.
- [60] A. Wojcik and G. Metze, Some relationships between Post and Boolean algebras, *Proc. 1971 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, New York.
- [61] M. Yoeli and G. Rosenfeld, Logical design of ternary switching circuits, *IEEE Trans. Computers*, EC-14 (1965) 19-29.

contents

List of Contributors	vii
Preface	ix
Chapter 1. Introduction, foundations, history, motivation	2
Part I. Algebraic Theory	
Chapter 2. From fixed to mixed radix	15
Chapter 3. Decisive implication	71
Chapter 4. Post algebras through P_0 and P_1 lattices	115
Chapter 5. An algorithm for axiomatizing every finite logic	137
Chapter 6. Completeness properties of multiple-valued logic algebras	144
Part II. Logic Design and Switching Theory	
Chapter 7. Computer simplification of multi-valued switching functions	189
Chapter 8. Minimization of multivalued functions	221
Chapter 9. The Allen-Givone implementation oriented algebra	262
Chapter 10. Multi-valued asynchronous networks	283
Chapter 11. Vector Boolean algebra and vector Boolean calculus; Theory and applications	308
Chapter 12. Binary network hazard elimination by multi-valued gates; Techniques and relative costs	339
Part III. Threshold Logic Design	
Chapter 13. A monograph on ternary threshold logic	355
Part IV. Physical Components and Implementations	
Chapter 14. Electronic circuits for multi-valued digital systems	397
Chapter 15. Multiple-valued negative resistance integrated circuits	420
Part V. Applications	
Chapter 16. Application of multiple-valued logic: possibilities and problems	475
Chapter 17. Multi-valued logic in arithmetic units	485

05200007

Chapter 18. Variable-valued logic and its applications to pattern recognition and machine learning	506
Chapter 19. Multiple-real-valued Walsh functions	535

introduction

chapter 1

an introduction to multiple-valued logic

david c. rine

Computer scientists, computer engineers, applied mathematicians and physicists are familiar with options in which there are no middle choices between true and false. Statisticians are familiar with the soft logic of probability, and physicists are familiar with the fuzzy logic of uncertainty. The lack of such choices is inconvenient and even critical when trying to determine whether or not the status of a computer system is go, wait or no-go. Multiple-valued logic is concerned with these intermediate choices.

The development of multiple-valued logic as viewed by the authors mentioned within this book is contemporaneous with the computer age and is clearly related to computer science, where there is by default a well established connection of 2-valued logic to computer structures and programs. The emphasis herein is on the development of multiple-valued logic as interrelating with theoretical and applied work in areas of computer science such as switching theory, programming and architecture. Parts I and II of this book discuss switching theory and theories of algebra and logic, while Part V treats multiple-valued aspects of programming, Part III describes full scale implementations, emulated and actual and Part IV gives further developments concerning special implementations or applications as they arise in threshold logic.

The relation between Boolean algebra and computer science and computer engineering began in the early 1940's. This relation was established because of the construction of actual computers and an increasing number of models comparing logical, computing and living systems by engineers, physicists, applied mathematicians and scientists from all disciplines.

The development of multiple-valued logic in the twentieth century can be dated at around 1920. However, the first algebra of n -valued truth-functionally complete logic corresponding to the work of Post was not formulated until the early 1940's.

This relation between the development of multiple-valued logic in its modern era and computer science is not surprising in consideration of the above connection. Early investigators directed their work toward disjunctive normal forms since these forms correspond in an obvious way with Boolean logic truth-tables and some of the associated minimization techniques. It is not surprising therefore that a number of computer scientists in the eight years following 1955 described the normal form theorem for Post functions, and this led

to the identification of certain operations occurring in other non-classical logics which are of interest in computer programming and computer engineering.

Some early research reported the change of state in relays and relay circuits. The use of 4-valued logic in order to analyze transients or faults was considered in the early 1950's. But since the intermediate values of transients need not be specified in certain transient or fault analyses, other approaches which do not specify intermediate values were also investigated.

In 1958 the first full scale 3-valued computer was completed at Moscow State University in the Soviet Union. While this encouraged work on subsystems such as arithmetic units, the lower order programming language devised for this Russian computer was so difficult to use that there was little insight into 3-valued logic by users.

Although higher order languages such as Fortran and Algol were conceived in the mid-1950's, these languages incorporate only 2-valued logic explicitly, and so disguise the use of concepts from multiple-valued logic when they occur within programs using these languages. The only early exception is the "arithmetic if" statement of early Fortran, which was a three-way branch, and the "computed go to" statement. Besides the familiar 2-way branch conceived in 2-valued terms, the multiple-valued n -way branch, where $n \geq 2$, may be simply devised in n -valued terms, but in 2-valued terms only through consecutive levels of nested 2-way branches. A large number of such levels of nested branches increases psychological complexity which may be faced by some programmers when writing programs. Further, the arguments in logico-arithmetic expressions used in branching are multiple-valued due to the multiple arithmetic constants of the computer, while the range of such expressions is 2-valued due to the two logical constants of the computer.

A higher order programming language with multiple-valued branching was IBM's Mathematical Formula Translating System of 1954, Fortran I. This language contained a control statement called the "computed go to" which allowed unrestricted branching to any of a number of multiple program statements depending on the value of a positive integer valued argument. The Darnsdtadt Meeting of 1955 resulted in an international programming language known as Algol 58, an algorithmic language. This language has the same kind of branching capability as the computed go to. Although the computed go to was an improvement over the arithmetic if in Fortran and other statements yielding only 3-valued or 2-valued branching, it very often led through poor software design to terrific program complexities and intolerable bother. Due to these complexities, flow-charting projects could probably achieve only average code-design rates of two to three statements per man-day, because of the time wasted on debugging and recoding.

There were a number of improvements in software design during the five years following 1962. Multiple-valued branching was provided in other languages such as Lisp. Also, the go to concept was replaced with the idea of the "case clause" by