MICROCOMPUTER MANAGEMENT AND PROGRAMMING

CAROL ANNE OGDIN

MICROCOMPUTER MANAGEMENT AND PROGRAMMING

CAROL ANNE OGDIN

Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632

Library of Congress Cataloging in Publication Data

Ogdin, Carol Anne. (date)
Microcomputer management and programming.

Includes index.
1. Microcomputers. I. Title.
QA76.5.035 1980 001.6'4'04 79-18068
ISBN 0-13-580936-3

Editorial/production supervision by Marianne Baltzell
Interior design by Barbara Kelly
Cover design by Edsal Enterprises
Manufacturing buyer: Gordon Osbourne

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *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*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

MICROCOMPUTER MANAGEMENT AND PROGRAMMING

PREFACE

During the first decade of microcomputers, industry has turned most of its efforts to training digital logic designers in the rudiments of programming. Those designers have solved most of the easy problems, and their skills are being outstripped by the demands for new products with more capabilities. The complexity in software is reaching a crisis stage for the traditional designer.

The few interdisciplinarians who can function well in both digital logic design and software design are in constant demand, but the supply is severely limited. The largest remaining pool of talent that can be drawn upon for satisfying the apparently insatiable microcomputer business are among the systems and applications programmers and the systems analysts already dealing effectively with minicomputers and larger-scale EDP systems. Yet, those individuals have purposely remained aloof from the computer engineering problems and are ill-equipped to take on microcomputer projects.

Management is, as a consequence, perplexed. Technically sophisticated managers familiar with the vagaries of major software system implementation are now being asked to manage microcomputer projects, but those managers, too, are ill-equipped.

This book is aimed at those managers, programmers, and analysts. Their involvement with microcomputer projects, as users, evaluators, and implementors is virtually certain. As microcomputers continue their proliferation, it is clear that distributed computing networks, manufacturing and process control, and even traditional EDP problems, will become more and more dependent upon the widespread availability of cheap computing power. Furthermore, new opportunities for product development and added sophistication in existing manufactured goods will continue to be recognized. Those products will demand the sophisticated skills of knowledgable managers, programmers, and analysts.

All of the microcomputer literature has been oriented toward readers with a firm grasp of digital electronics technology and an ignorance of software. Little that has been written on micros is addressed to the reader who has a complete grasp of software design, but negligible experience with digital logic and computer design. The first instance, to my knowledge, was the combined November-December 1977 issue of *Mini-Micro Systems* which I wrote, and which formed the nucleus for this book. Within the limited space a magazine provides, I tried to show software experts the rudiments of computer and logic design. Although too brief to have been the basis for a complete understanding of these complex topics, the response to that level of treatment of the subject has motivated me to produce this more complete effort.

Microcomputers are more than cheap computers. They demand of the designer an awareness of both software and digital logic principles. The complete designer is an interdisciplinarian, but it takes about as long to become proficient at circuit design as it does to grow a good systems programmer. Few people have time to master both skills. Managers will find, then, that future complex microcomputer-based systems will be created by teams of designers, individual members of which will have specialized backgrounds in either electronics or computer science. The digital designers have learned enough software skills to be conversant with the problems; nearly all of them have acquired the ability to read and comprehend (if not create) computer programs. To balance the team it will remain for the software experts to acquire enough skills to be able to read and comprehend (if not create) circuit diagrams for microprocessor-based computer systems. When, at last, management has assembled a balanced team with combined hardware and software skills, some of the huge applications problems can be faced with reasonable certainty of success.

This book is written for managers who need to know enough about the technology to be able to understand and mediate some of the disputes that invariably arise among team members. It is intended to provide a matrix in which the manager can set individual decisions to be made so as to place them in proper perspective. Some of the chapters may be too detailed for a manager's needs, but a brief skimming of the chapters can offer some insights into designers' problems.

This book is also written for programmers and analysts who will be called upon to solve major applications problems with an unfamiliar technology. It is intended as a guide to getting started. More, it is intended as a means for destroying some of the software designer's dearest-held myths about how computers and the relevant software should be designed. Some of the chapters may be too oriented to the larger issues of management of these kinds of projects, but they are worth skimming to find out what problems other team members may have to face.

Microcomputers have come at a time when software engineering is emerging as a discipline—and none too soon. Just as some of us were becoming weary of constructing ever-larger and ever more unreliable software systems, microcomputers have come along as a major simplifying influence on good design practice. This is probably the most exciting time in history to be in the computer industry. I hope you, too, will find the joy I feel in creating a new computer to solve a problem never before broached.

Carol Anne Ogdin
Alexandria, Virginia

CONTENTS

two
MICRO APPLICATIONS 22

three
COMPUTER ARCHITECTURE 50

four
MICRO SELECTION 67

five
STORAGE AND INPUT/OUTPUT 106

six
MICROCOMPUTER PROGRAMMING 161

seven
INTERFACE TECHNIQUES 197

eight
HARDWARE/SOFTWARE TRADEOFF 239

nine
IMPLEMENTATION AND CHECKOUT 253

eleven

PROJECT PLANNING 323

appendix

UNDERSTANDING THE LANGUAGE OF ELECTRONICS 335

INDEX 343

one

THE TECHNOLOGY

The prospect of $25 computers captures the imagination of clever system designers. The potential uses are so pervasive that new applications are being announced more quickly than most designers ever thought possible. Following the trends of the decades, from multimillion-dollar monsters, through $10,000 table-top minis, and down to $1,000 computer-on-a-card micros, technology will continue the downward spiral in computer cost. The result will be to make the computer an even more important element in our daily lives. There is an instant of time—now—when specialized computers may cost under $20 in certain uses. Just as those one-chip computers rival the power of the digital computers of the 1940s, the future may reduce the power in today's large-scale data processing systems to mere hand-held devices, and the cost may be a mere $20. Today's microcomputer is just another point on a continuing trend, but this point must be understood in order to be able to exploit technology now and in the future.

The potential for microcomputer applications has already outstripped even the most inventive minds. Systems designers have found more new uses for micros than any semiconductor vendor ever dreamed possible. Each of those new ideas stimulates another inventive designer, and the end of the computer era gets pushed further and further into the future.

But, what is a micro? How is it used? Is it merely a cheap replacement for a conventional computer? Can designers wait for the technology and the software support to mature? Indeed, the advent of the micro poses more questions than it answers. While new applications abound, the shape of complex questions that would have been irrelevant 10 years ago is becoming clear. Will arrays of thousands of micros all connected together form the basis of architecture in the future? It's possible.

While a micro may be treated simplistically as "just another computer," it is foolhardy to slavishly retain that notion. The mind boggles at the concepts that microcomputers permit designers to consider as potential and practical. The facts are, though, that the micro is being used successfully by thousands of companies as just another system component. Those designers are finding that the micro has opened up wider opportunities for profitable ventures, but at the cost of changing traditional techniques and at the cost of learning more about the fundamentals of computers themselves.

Industrial and professional periodicals now carry dozens of articles on microcomputers and their applications each month. Most of these assume the reader knows digital circuitry and is proficient in programming. But in fact, most system designers have little direct experience with the electronics inside the hardware, and this is especially true of designers who emerge from a systems software background. These professionals are accustomed to fitting together compatible central processing units (CPUs), memories, and peripherals in order to solve a problem. Problems that can't be solved with off-the-shelf building blocks are often deemed "unsolvable." These designers will find that some change in their behavior will be essential in the coming years.

EVOLUTION OF THE MICRO

The classic diagram of a computer (Figure 1-1) is still relevant, after all these years. Even the ENIAC, conceived in the 1940s, was based on this concept, as are nearly all of the most modern computers. The Control and Arithmetic-Logic Unit (ALU) are often treated as indivisible, and that combination is called the Central Processing Unit (CPU). The CPU is central to the computer's flow of data: from input devices, through storage media, and to output devices. The intrinsic power of each of these functional boxes is what distinguishes the various classes of computers. As technology has progressed, what used to be a large-

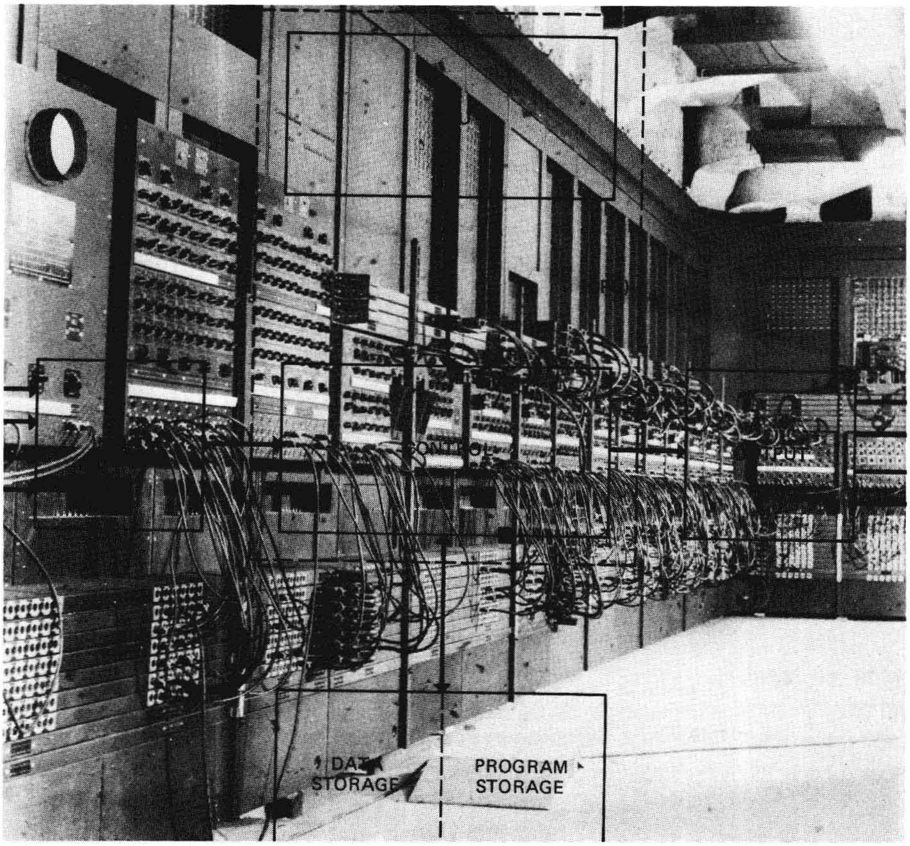

Figure 1-1 Even ENIAC, the first electronic digital computer, relied upon the classic model of all computers. (Photo courtesy of Smithsonian Institution, Photo No. 61 699A.)

scale computer has been replaced by the mini, and micros are now supplanting minis in many applications.

The genesis of this new style of computer—the micro—is well understood now, in retrospect, although few people could imagine it before the fact. While the semiconductor technology was making tremendous leaps in the amount of complexity that could be installed in a single low-cost package, computer architectures were being more and more simplified to meet the challenges of nonnumeric problems like control. During the early 1970s, these two trends crossed one another, and the computer-on-a-chip was born (Figure 1–2).

In the early days of computers, the central processing unit was made up of bank upon bank of vacuum tubes. The reliability was measured in hours between failures, and the air-conditioning plant was often larger than the computer itself. Typical of the technology of that era was a single accumulator (Figure 1–3), held by two of the original ENIAC

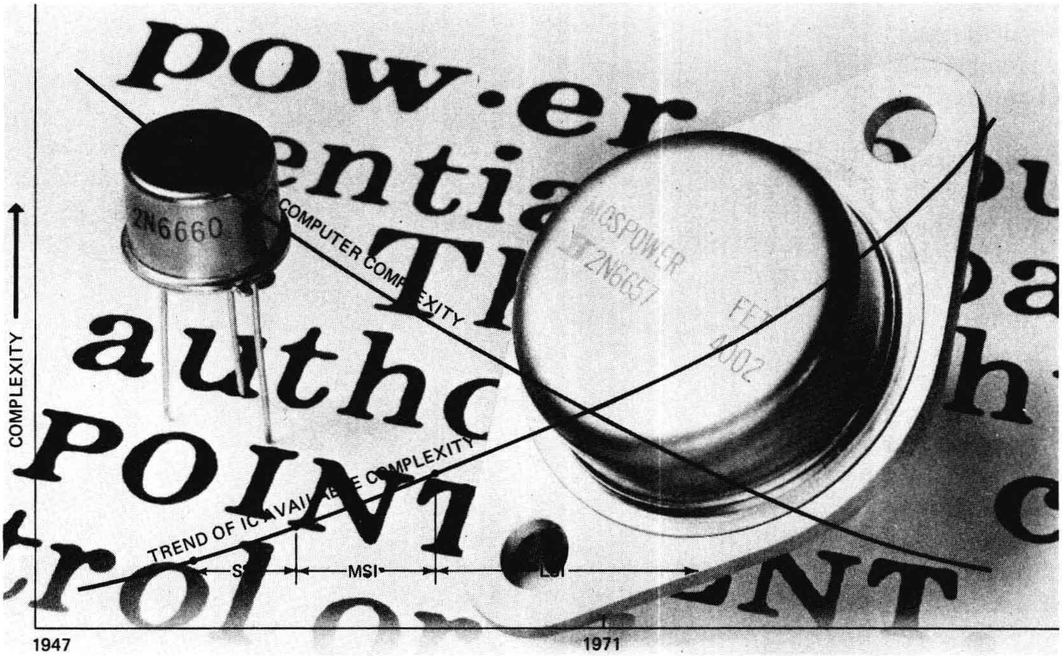

Figure 1-2 The creation of the transistor led to reliable computers; while more complex devices became available, computers were declining in complexity. (Photo courtesy of Siliconix.)

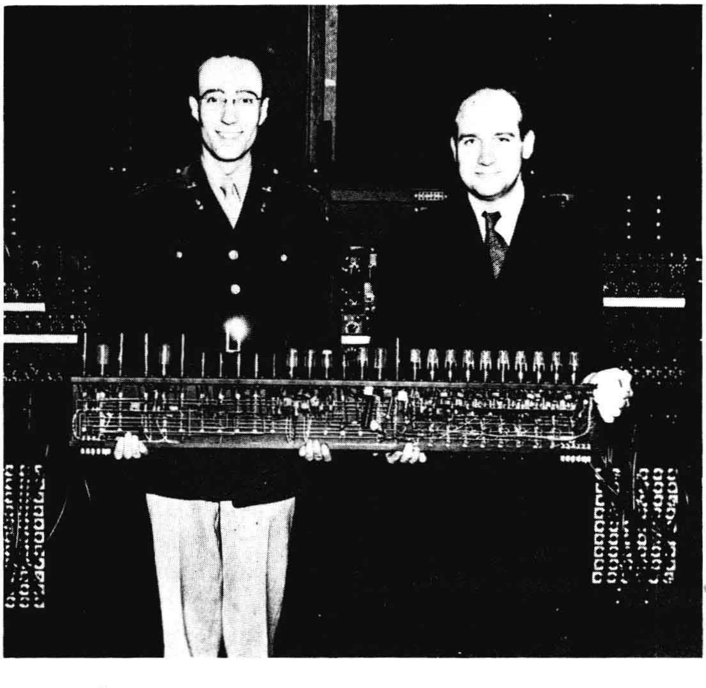

Figure 1-3 Two of ENIAC's sponsors hold a single accumulator. (Photo courtesy of Smithsonian Institution.)