Introduction to Microprocessors: Software, Hardware, Programming

LANCE A. LEVENTHAL

Introduction to Microprocessors: Software, Hardware, Programming

Lance A. Leventhal

Engineering and Technology Department
Grossmont College

PRENTICE-HALL, INC., Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

LEVENTHAL, LANCE A (date)
Introduction to microprocessors.

Includes bibliographies.
1. Microprocessors. I. Title.
QA76.5.L486 001.6'4'04 78-7800
ISBN 0-13-487868-X

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*

Preface

Microprocessors have already had far-reaching effects in almost every industry. New processors, new hardware and software intended for use with processors, and new products that incorporate microprocessors seem to appear daily. As usual, textbooks and other educational materials have lagged far behind the spread of the technology itself. The manuals written for specific microprocessors are intended as references for users rather than as textbooks; articles in the trade press or technical journals discuss limited subjects or particular applications. This book is a general introduction to microprocessors for advanced undergraduate students; the students are assumed to have had introductory courses in programming and digital circuits. This textbook should prove useful and interesting to undergraduate and graduate students in electrical engineering, other engineering disciplines, the physical and natural sciences, mathematics, computer science, and the health sciences. The book will also serve the needs of engineers, technicians, administrators, programmers, teachers, and others who need a self-study or reference text on microprocessors.

A major problem in teaching about microprocessors is that they cannot be understood solely from a software or a hardware point of view. Programmers will find that microprocessors have architectures, instruction sets, and software much like other computers with which they may be familiar. Engineers will find that microprocessors have the same physical properties, input and output signals, and operating characteristics as other integrated circuits with which they may be familiar. However, both programmers and engineers will discover that developing systems based on microprocessors requires understanding of both hardware and software. I have taken the approach in this book that the use of microprocessors does involve a shift of the design burden from hardware to software and I have, therefore, emphasized software. At the same time, I have discussed the hardware aspects of microprocessors at some length with particular emphasis on the design of memory sections, the use of standard and specialized integrated circuits, and the interfacing of simple peripherals. I have not discussed digital design since I believe that several textbooks adequately cover that subject at various levels. Thus I have written a software-oriented book while not ignoring the associated hardware problems.

Rather than surveying the entire spectrum of microprocessors or inventing an example for study, I have chosen to focus on the two most widely used devices, the Intel 8080 and the Motorola 6800. These processors are similar in performance to most of the standard devices on the market. They are also sufficiently different from each other, both in programming and interfacing techniques, to be representative of the entire range of devices. I believe that adequate surveys of microprocessors are available and that a textbook should provide a detailed treatment of one or two devices. I have based my choice on the choice of the marketplace; I see no evidence that these two processors (or any others) are superior to their competitors. This book does occasionally mention some of the special features of the various competing processors.

The basic organization of the book is as follows:

Chapter 1 introduces the subject of microprocessors. It compares microprocessors to minicomputers and large computers as well as to other large-scale integrated circuitry. It then describes semiconductor technologies and memories, discusses the advantages and disadvantages of microprocessors as compared to the competing design techniques, presents the areas in which microprocessors have been applied, and gives some specific examples of microprocessor-based products.

Chapter 2 considers the architecture of microprocessors. The various sections of a computer—central processing unit (CPU), memory, and input/output—are briefly discussed. The rest of the chapter concentrates on the CPU, describing the registers, arithmetic unit, and instruction-decoding mechanism. The final section presents the architectures of the Intel 8080 and Motorola 6800.

Chapter 3 describes instruction sets. The first sections contain a general description of instruction formats, addressing methods, and types of instructions. The last section describes the instruction sets of the Intel 8080 and Motorola 6800.

Chapter 4 deals with assemblers. The first sections discuss the advantages and disadvantages of various language levels and the general features of assemblers. The last section is a specific description of the standard Intel 8080 and Motorola 6800 assemblers.

Chapter 5 discusses assembly language programming for the Intel 8080 and Motorola 6800 processors. The chapter begins with simple programs and proceeds through loops, string and character manipulation, code conversions, arithmetic, lists and tables, and subroutines.

Chapter 6 considers the entire software development process. It describes problem definition, program design, coding, debugging, testing, documentation, and maintenance and redesign. The last section briefly discusses development systems.

Chapter 7 deals with the memory section. It describes the basic interaction between the microprocessor and memory. It goes on to consider the interfacing of simple memory sections and the design of busing structures required for more complex sections. The chapter then describes the design of memory sections for the Intel 8080 and Motorola 6800 and the operation of their instruction cycles.

Chapter 8 discusses input/output. It begins with a description of input and output procedures. It then considers simple input/output sections and more complex input/output sections which require a busing structure. The chapter then discusses some circuits that are widely used in I/O sections and some simple I/O devices. The final part of the chapter describes the specific hardware and software required to interface simple I/O devices to the Intel 8080 and Motorola 6800 microprocessors.

Chapter 9 presents interrupts. It describes the uses, advantages, and disadvantages of interrupts. It then discusses interrupt procedures, the features of interrupt systems, and the handling of specific sources of interrupts. The chapter next describes the programming and interfacing of interrupt-based systems using the Intel 8080 and Motorola 6800. The chapter concludes with a brief consideration of direct memory access.

The Appendices present background material, tables of codes, and instruction sets. Appendix 1 deals with the binary number system, Appendix 2 with logical functions, Appendix 3 with numerical and character codes, Appendix 4 with semiconductor technologies, and Appendix 5 with semiconductor memories. Appendices 6 and 7 contain the instruction sets of the Intel 8080 and Motorola 6800 microprocessors. The Appendices are followed by an extensive Glossary.

Obviously, I have omitted many subjects. I have described the purpose of microprogramming without describing the procedures for microprogramming either at the designer or user level. The microprogramming of microprocessors, in itself, probably deserves an entire book. The use of high-level languages has been explained, but no specific examples have been presented. Such techniques as multiprocessing, pipelining, parallel processing, and virtual memory have also been omitted. Microprocessor peripherals and the wide variety of associated digital and analog integrated circuitry that is currently available have been described sparingly. Direct memory access has received only a brief treatment. Some of those topics may be dealt with at a later date.

I have attempted throughout this textbook to illustrate various topics by presenting examples of the actual use of microprocessors. I have tried to develop these examples in a logical manner so that the reader can apply them directly to laboratory or engineering problems. In the hardware area, I have used standard symbols, design techniques, circuit elements, and documentation. In the software area, I have used standard flowcharting symbols, and have structured and documented all programs in accordance with the methods recommended by modern theorists. I have tried not only to present useful examples but also to show the rules of structure and documentation which I believe to be vital to the efficient development of microprocessor-based systems.

A key difficulty in writing a book about a new technology is the rapid obsolescence of information. Obviously the time required to publish and distribute this textbook will make some of its information obsolete. I have not tried to predict the future course of microprocessor technology, but I have tried to focus on processors that are widely available, have been used in significant applications, have considerable hardware and software support, and will be important factors in the marketplace for years to come. I have also attempted to present continuing trends, current problems, and promising new approaches. However, I realize that readers will have to obtain up-to-date information from the manufacturers' manuals and technical notes, articles in trade and scholarly journals, and special conferences and short courses. I have described the sources of such information and have presented basic approaches to microprocessors that will enable the user to take maximum advantage of these sources.

The field of microprocessors is clearly in a state of rapid expansion. More courses, articles, and conference sessions on this subject appear all the time. We are just beginning to see the results of this new technology in actual products. The capacity for intelligent and flexible control of everyday systems is now available at a reasonable price—the use of this control is a challenge that is open to all the readers of this book.

Acknowledgments

My special thanks go to Mr. William Tester of Grossmont College, who was responsible for introducing microprocessors into the Grossmont College curriculum and for providing much of the support which made this book possible. I would also like to thank Mr. Colin Walsh of Grossmont College for his continuing assistance and Mrs. Teddy Ferguson of Grossmont College for her typing and proofreading help. Others who have assisted me include Mr. David Bulman and Mrs. Kati Bulman of Pragmatics, Mr. Karl Amatneek of KVA Associates, Mr. Ken Reider and Mr. Tony Earle of Earle Associates, Mr. Jeffrey Haight of Data/Ware Development, Professors Fredric Harris and Nicholas Panos of San Diego State University, Mr. James Gordon and Mr. Franklin Antonio of Linkabit, Mr. Terry Benson of Intel, Dr. James Tiernan of Linkabit, Mr. Romeo Favreau of Sorrento Valley Group, and Dr. Donald Rauch of Sysdyne. Mr. Stanley Rogers of the Society for Computer Simulation deserves recognition for prodding me constantly to improve my writing style. My wife, Donna, has been patient and understanding. I, of course, bear full responsibility for the contents and organization of this book but all of those whom I have mentioned and others besides should share in any credit.

LANCE A. LEVENTHAL

Solana Beach, California

Contents

APPENDICES

1 Introduction to Microprocessors

1.1 COMPUTING AND MICROPROCESSORS

The brief history of computers is marked by rapid advances in technology and in the number and variety of applications. The first computers, built in the late 1940s and early 1950s, were used to solve complex scientific problems. Today computers that are far more powerful than the early machines are used in such mundane applications as electronic games, cash registers, scales, calculators, and household appliances. An individual can purchase a computer like the one shown in Fig. 1.1 for a few hundred dollars and use it in a variety of home projects.

Figure 1.1 A home computer (courtesy of MITS, Inc.).

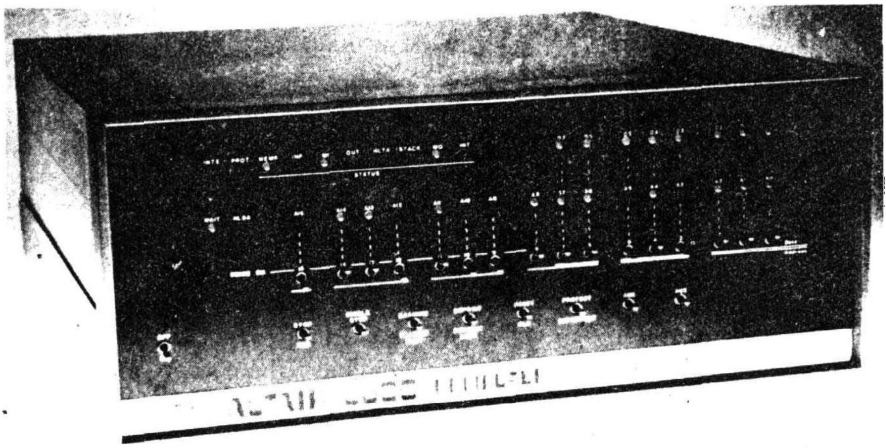
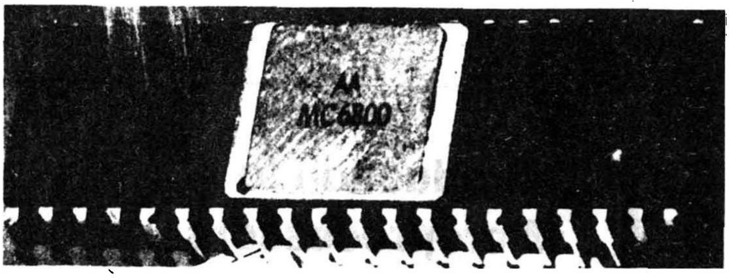


Figure 1.2 A typical microprocessor (courtesy of Motorola Semiconductor Products, Inc.).



The latest development in computer technology is the *microprocessor*, a device that has all the functions of the central processing unit (CPU) of a computer on one or a few tiny pieces of silicon. Figure 1.2 shows a typical microprocessor. Such a device can fetch instructions from a memory, decode and execute them, perform arithmetic and logical operations, accept data from input devices, and send results to output devices. A microprocessor, together with a memory and input/output channels for communication with the outside world, forms a full-fledged computer or *microcomputer*. Simple microcomputers cost as little as $10. Complete *microcomputer systems*, with a chassis, front panel, and power supply in addition to the microcomputer, cost less than a thousand dollars and have as much processing power as giant computers of the 1950s that cost hundreds of times that amount. The rapidly decreasing cost of computers contrasts sharply with the rapidly increasing cost of most other items.

The microprocessor is a continuation of the trend toward smaller computers that began in the middle 1960s. In the early years of computer development the emphasis was on larger and more powerful machines. Computers were so expensive that only large institutions could own them and only specially trained personnel could operate them. New technologies, such as transistors and integrated circuits, provided greater speed but did not reduce costs. Computers remained remote and mysterious objects.

Minicomputers started the present trend. The first minicomputers were primitive by computer standards and still cost tens of thousands of dollars. Nevertheless, laboratories, factories, and smaller institutions that could not afford large computers could now purchase a small computer like the Digital Equipment PDP-8, Data General Nova, Scientific Data Systems 92, or IBM 1130. Cheaper electronic circuits led to cheaper minicomputers; by 1970 a small minicomputer for use in a laboratory, office, factory, warehouse, or classroom cost a few thousand dollars.

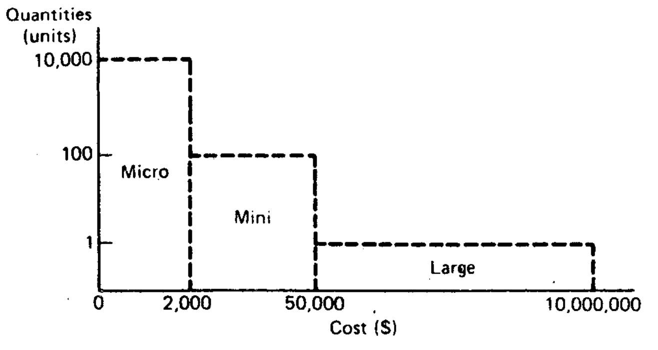
Yet developments in integrated circuits went far beyond reducing the cost of minicomputers. Soon the construction of a single integrated circuit that could perform the functions of a computer became possible. Such complex circuits (called *large-scale integration* or *LSI*) can be manufactured by the thousands at costs not much greater than those of simple circuits. The cheap computer, long a favorite dream of science fiction writers, thus became a reality. Manufacturers have already produced more microprocessors than all other computers combined. By 1977, complete home computers including a keyboard, video display, and cassette recorder were available for under a thousand dollars.

Besides comparing microcomputers to other computers, the rest of this chapter describes the origin and history of microprocessors, the characteristics of the semiconductor technologies from which microprocessors are produced, the features that microprocessors share with other integrated circuits, the semiconductor memories with which microprocessors are used, and, finally, some applications of microprocessors.

1.2 LARGE AND SMALL COMPUTERS

Cheap computers have made many new computer applications possible, a trend started by the minicomputer and extended by the microprocessor. Figure 1.3 shows the costs of microcomputers, minicomputers, and large computers, along with typical quantities that a user might purchase at a time.

Figure 1.3 Computer costs and typical quantities.



Large general-purpose computers like the IBM 370, Univac 1100, or Burroughs 6700 serve two major functions:

1. Solving complex scientific and engineering problems, such as spacecraft guidance, weather prediction, or electronic and structural design.

2. Performing large-scale data processing, such as the handling of records for banks, insurance companies, stores, utilities, and government agencies.

These tasks involve extremely large numbers of calculations or data transfers. Solving typical scientific problems requires the solution of complicated equations that could not be solved by hand. Handling business problems requires the processing of large numbers of records and the handling of large amounts of peripheral input and output.

Minicomputers and microprocessors will not replace large computers in such applications. Small computers can, of course, solve similar problems when the calculations are less complex or the amounts of data are smaller. Thus a minicomputer or microprocessor could perform laboratory calculations or handle records for a small business. Nevertheless, the greatest usage of mini and microcomputers has occurred in areas outside the typical applications of large computers.

Normal applications for mini and microcomputers have the following characteristics.

1. The computer is a system component. The overall system, which might be a piece of test equipment, a machine tool, or a banking terminal, uses the small computer much as it might a switch, power supply, or display. The computer may not even be visible from the outside.

2. The computer performs a specific task for a single system. It is not shared by different users as a large computer is. Instead the small computer is part of a particular unit, such as a medical instrument, typesetter, or factory machine.

3. The computer has a fixed program that is rarely changed. Unlike a large computer, which may solve a variety of business and engineering problems, most small computers perform a single set of tasks, such as monitoring a security system, producing graphic displays, or bending sheets of metal. Programs are often stored in a permanent medium or read-only memory.

4. The computer often performs real-time tasks in which it must obtain the answers at a particular time so as to satisfy system needs. Such applications include machine tools that must turn the cutter at the right time in order to obtain the correct pattern or missile guidance where the computer must apply thrust at the proper time in order to achieve the desired trajectory.

5. The computer performs control tasks rather than arithmetic or data processing. Its primary function might be managing a warehouse, controlling a transit system, or monitoring the condition of a patient.

Using minicomputers and microcomputers differs from using large computers. The user of a large computer system writes programs in a convenient language (such as FORTRAN, BASIC, or PL/I), submits them to the computer center staff, and receives results in printed form or on magnetic tape or other medium. The user can take advantage of a variety of peripheral devices, standard programs, and other features of a large computer center. The user need not worry about how the computer operates or how it communicates with memory and input/output devices.

Those who wish to use minicomputers or microcomputers as system components will find the situation quite different. Minicomputers and microcomputers rarely have the software and peripherals needed for convenient use. Programming small computers is tedious and time-consuming. Furthermore, the interaction of the computer with the total system is a critical problem. The designer must understand the details of how the computer works in order to use it effectively.

Small computer applications thus involve interrelated hardware and software. The designer must divide tasks between hardware and software, considering both cost and speed, and write programs to obtain data from input devices and send results to output devices. Timing considerations are critical in both hardware and software design.

The small computer that goes into the final product is generally not suitable for use in the development stage. Programs for large computers are developed on the same system on which they are used. Small computers, however, that are part of an instrument or machine tool will only have enough memory, peripherals, and software to perform their functions; anything extra would increase system cost without improving performance. As a result, such computers will not have card readers or line printers to handle input/output, compilers or debugging packages to simplify program development, or tape or disk systems to store programs or data. The computer that goes into the system can handle system functions—not system development.

Special equipment (called a *development system*) is therefore used during the development stage. This equipment may involve the same computer (*self-development system*) or other computers (*simulator* or *cross-development system*). A typical development system has peripherals for entering programs and data and recording results, a fairly large memory for holding user and system programs, software and hardware features useful in writing and debugging programs, and mass storage (disk or tape) for saving programs so that they can easily be retrieved. Such development systems include software, hardware, peripherals, and interfaces that have nothing to do with the ultimate product; they merely make designing easier. The task of developing programs, even when the programs are ultimately intended for mini or microcomputers, is better suited to large computers. A program that translates FORTRAN to machine language, for instance, can run much faster on a large computer than on a small one and can use peripherals that are part of the large system. The computer that does the translation need not be the one for which the program is intended.

The user of minicomputers or microcomputers will thus find system development quite awkward. Programs are often written in a very different environment from that in which they will ultimately run. Frequently, hardware and interfaces are developed concurrently with programs, and the user faces a difficult task of system integration. The programs must often perform complex input/output operations with real-time constraints.

1.3 COMPARISON OF TYPICAL COMPUTERS

Table 1.1 compares a large general-purpose computer, a large minicomputer, a small minicomputer, and a microcomputer. The large computer described is an IBM 370/Model 168 (see Fig. 1.4), a computer widely used for data processing. The large minicomputer described is the Digital Equipment (DEC) PDP 11/45 (see Fig. 1.5). The small minicomputer is the Computer Automation NAKED MINI (see Fig. 1.6), a popular system computer. The microcomputer is the Intel MCS-80, based on the widely used Intel 8080 microprocessor (see Fig. 1.7). Other manufac-