ELEMENTARY ASSEMBLER LANGUAGE PROGRAMMING

FLOYD E. HAUPT

ELEMENTARY ASSEMBLER LANGUAGE PROGRAMMING

FLOYD E. HAUPT
Brigham Young University

CHARLES E. MERRILL PUBLISHING COMPANY
A BELL AND HOWELL COMPANY
COLUMBUS, OHIO 43216

International Standard Book Number: 0-675-09158-6

Library of Congress Catalog Card Number: 75-178676

1 2 3 4 5 6 7 8 9 10—80 79 78 77 76 75 74 73 72

Printed in the United States of America

ELEMENTARY ASSEMBLER LANGUAGE PROGRAMMING

To my wife Marian,
and my children:
Ruth, Lois, Joyce, Carl, Robert, and Patty

Preface

This book has been written for those who want to understand some of the basic concepts involved in programming a computer on the machine and assembly language levels. Some people will find that understanding such points as the difference between Real and Integer forms of data will aid or extend their mastery of problem-oriented languages, such as FORTRAN and BASIC. Others will discover a wonderful new world in the field of programming, a field for which they have a natural talent.

We are also interested in students who are not science-oriented, but we do have some prerequisites in mind. The student will receive the greatest benefit from this book if he has had at least 2½ years of high school mathematics, including trigonometry. Students with a little less mathematics may do quite well, but the teacher should make appropriate adjustments in the course.

Although the book concentrates on helping students, the teachers will find that their problems have not been ignored. The large number of Exercises and actual computer runs in the text and the associated *Workbook* make it possible to create a course as independent of computer facilities as is desired. Neither computer "downtime" nor lack of a budget can interfere seriously. The use of an actual computer is strongly recommended, however, and Exercises have been provided for those who have such facilities.

One good reason for the popularity of such languages as FORTRAN and BASIC is that, to a large extent, they are machine independent. We have adopted this advantage in teaching machine and assembly languages

by defining a minimachine and simulating it on the widely available IBM 360. The software package is available from the publisher. Most classes contain students who grasp the ideas quickly and then seek more advanced projects. A good assignment for such students would be to have them write a simulator for your particular machine.

We have chosen a language with 64 instructions and a memory size of 512 words. The small memory size allows a dump of the entire memory on a single page. Since the ideal word size seemed to be 18 bits, we have used an octal representation of machine words. The change to hexadecimal is easy once the student understands how other number bases can help him. We have included some Exercises of a hexadecimal type to aid in the transition, if it is necessary. However, manufacturers are now making 9-bit units, which may encourage the use of octal notation.

By stressing the ideas of sets and functions, we have given the teacher another good way to illustrate the value of the "new math." And, by using elementary concepts of linguistics, we have not only introduced the student to an important new subject but have also made the computer concepts more understandable.

The textbook bears the name of a single author, but he must admit that the ideas contained in it are not his alone. A large part of the credit must be given to Bernard N. Daines, who wrote the simulator, and to Ronald M. Davis, who had—and pushed—the original idea.

Floyd E. Haupt

Provo, Utah

Contents

1

The Computer World

1.1 Introduction

Computers are here to stay, and almost everyone is glad. This is not to say that modern computers are trouble-free; everyone has heard stories about futile arguments with computerized credit departments. The approach in this text, however, will be in a positive vein. Few people would want to lose the many advantages of modern technology merely to eliminate some difficulties.

1.2 A Brief History of Computers

The first operative computer probably consisted of a fistful of fingers. This model has been so reliable that it is used even today. While toes were no doubt of equal importance in the early days, the invention of shoes has made the toe model obsolescent, if not obsolete. Man has never managed to overcome this problem of obsolescence. Rather, progress has magnified the problem to such an extent that a modern computer may become obsolescent sometime between completion of the design and production of an actual piece of hardware.

Some other early computers have survived progress. For example, the Chinese abacus, which consists of beads on strings in a framework, is used extensively today in the Orient. Skilled operators have beaten men trained on desk calculators, and for short problems the abacus can beat

an electronic computer, because even the fastest computer must rely on human operators. In some big computer installations the local humorist may have a glass-enclosed abacus sitting on the main console—with instructions to break the glass in an emergency. Obviously, the control of reliability is another problem man has not yet completely solved in the electronic age.

Another example of computer survival is the slide rule. The invention of logarithms by John Napier (1550–1617) made slide rules possible, because a slide rule is nothing but logarithms on a stick. The portability of slide rules makes them handy for quick calculations of limited accuracy. Engineers will continue to use them in the forseeable future.

But the survival of old style computing devices does not mean that they can handle all modern needs. Computational requirements during World War II hastened the design and production of large-scale computers. Although the first ones were primitive by today's standards, they were regarded as marvels. Like early models of the automobile, these early "large-scale" computers were cantankerous, unreliable, and hard to maintain. But they did achieve something that the Mia (M–1A) computer in the Li'l Abner comic strip sought: they were loved. In fact, they were used even after newer and better machines were made. (Loyalty is a great virtue, but it can be overdone!)

The electro-mechanical memory units of the early computers quickly gave way to improved devices of many types, which in turn have been replaced. Magnetic cores are in common use today, although other devices, such as thin films, are being introduced.

The art of telling a computer what to do is called *programming*. This, too, has been vastly improved during recent decades. Originally, the data and the program were separate entities. For example, the old IBM 602-A computer put the data into punched cards, but the program consisted of making electrical connections by means of wires plugged into a board. It was not long, however, until the CPC (Card Programmed Calculator) was introduced. In this device the program and data could be alternated in successive punched cards. But the CPC was shortlived: improved computers had the ability to store the data and program instructions in expanded electro-magnetic memories. Furthermore, new input and output devices were developed. Although some modern computers still use punched card input, others use magnetic tapes, magnetic discs, and magnetic inks (see any bank check), as well as other devices. And nobody knows with certainty what will be popular a few decades from now.

Computers of all eras may be classified as *digital* or *analog* devices. Analog computers use physical measurements of some type. Some of them are easy and fast to use, while others will tax the user's skill, but all

analog computers are quite limited in accuracy because the measuring device is some type of scale with physical subdivisions. For instance, a meterstick has centimeter subdivisions that are easy to see physically, but smaller subdivisions become increasingly hard—or impossible—to read. A digital computer, on the other hand, is essentially opposite to an analog machine. A digital computer's accuracy is limited only by the designer's decision to stop adding significant figures. For instance, a desk calculator could be designed to have as many digits per register as desired. Since each digit is represented by a printed numeral of the same size, there is no problem inherent in reading all of the digits in a number. The miniature computer to be described in this book is of the digital type.

Computers may also be classed as *general purpose* or as *specific purpose* machines. The first modern computers tended to be for specific purposes, but the many advantages of a general purpose computer became so apparent that the use of special purpose devices is now limited to such things as permanent machine controls. The idea of a general purpose computer is not new. The early machines had their problems, not because of faulty mathematical concepts, but because the technology of the day was incapable of producing a complicated yet workable machine. Charles Babbage, for example, worked on calculating machines from 1823 to 1842 before abandoning the project. Even the relatively simple electric desk calculators of today are both new and old: they are new in design but old in concept.

Exercises

1. Which of the following are analog and which are digital devices?
   a. A modern desk calculator
   b. An abacus
   c. A slide rule
   d. A gas station pump
   e. An automobile odometer
   f. An automobile speedometer
2. What is the difference between *obsolescent* and *obsolete?*
3. Use at least one modern encyclopedia to write a short report on each of the following items.
   a. The Mark I computer
   b. The ENIAC computer
   c. The EDSAC computer
   d. The BINAC computer
   e. The UNIVAC computer
   f. Herman Hollerith
   g. The binary number system
   h. Input and output devices
   i. Memory storage devices
   j. Computer languages
   k. Computer hardware and software
   l. John von Neumann
4. Build a simple abacus and learn to calculate with it.

5. Learn to multiply and divide with a slide rule. (*Hint: Collier's Encyclopedia* has a good description of a slide rule.)
6. a. How could three ammeters (current measuring devices) and two rheostats (current variation devices) be connected so as to record the addition of two numbers? Assume that a constant voltage is available. (*Hint:* Kirchoff's current law says that if no electricity accumulates at a given point in an electrical circuit, then the sum of the currents flowing into that point will equal the sum of the currents flowing out of the point.)
   b. Is this an analog or a digital device?

2

Basic Concepts of the Minicomputer

2.1 Introduction

There are two possible approaches to teaching the fundamentals of computing with a small but versatile computer. First, an actual miniature computer could be built. Second, a large-scale computer could be programmed so that a portion of it simulates a smaller and simpler, but actually operable, machine. This second approach has been employed here. A special program has been written to simulate this minimachine on the particular computer which the student has available. This assumes, of course, that one is using a machine for which a simulator has been written. Although the examples make use of a fictitious computer, the principles are basically the same as those used with a real machine.

Since hands-on experience is such a valuable teaching device, it is important for the student to obtain actual programming experience as soon as possible. A basic programming format will be introduced in this chapter and explained more fully in future chapters. In the meantime, elementary variations of the basic program are possible for the beginner. The novice should not be alarmed by the relatively little he understands about the programming language. Ignorance should motivate the study of some essential preliminary ideas in the next two chapters. Furthermore, even experienced programmers must follow mathematical or other instructions that they do not fully understand. A machine is no respecter of persons, hence *success depends on complete and exact obedience to rules.* Troubles will soon make apparent the truth of this statement.

If one does not have computer facilities available, he must depend entirely on the *Workbook* which supplements this text. However, the availability of a computer does not mean that the *Workbook* should be abandoned. It will provide time-saving illustrations and drill on basic ideas.

2.2 The Computer's Memory Unit

The memory of a computer may be likened to the warehouse of a business. It is a storage place for valuable items. Obviously, quick access to a stored item is very convenient. A businessman must know the location of items in his warehouse and keep track of how many of each item he has. He finds a system of inventory control highly desirable. Perhaps he simply paints two marks down the length of his warehouse floor to define an aisle and marks off adjacent rectangular areas which he numbers. Thus, his warehouse floor would look something like Fig. 2.2–1.

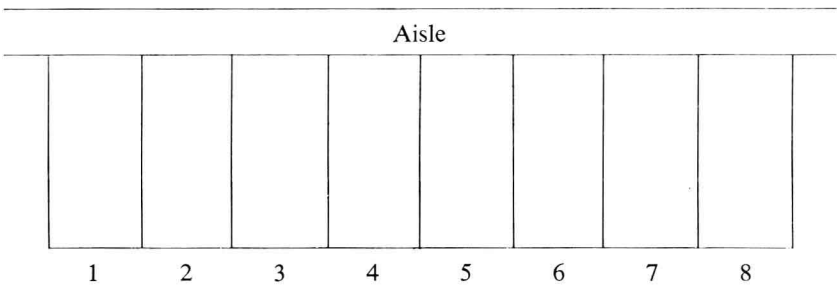

Figure 2.2–1

The aisle is kept free of stored items so as to provide easy access to each storage area. Also, the contents of each rectangle are arranged neatly so that the items can be counted quickly.

Suppose that Jack, the well-organized businessman, writes a large *J* on each case as his trademark. He keeps track of different batches of his product by storing them on separate rectangles. At a certain time he walks down the aisle and observes *J*-labeled boxes in the arrangement of Fig. 2.2–2.

Walking down the aisle is inconvenient—especially if Jack wants a quick answer to an inventory question—so he decides to build a box with eight slots and to keep a slip of paper, with the current number of cases recorded on it, in each slot. Each slot is numbered to correspond to a storage rectangle. Thus, at the moment, his handy box looks like Fig. 2.2–3.