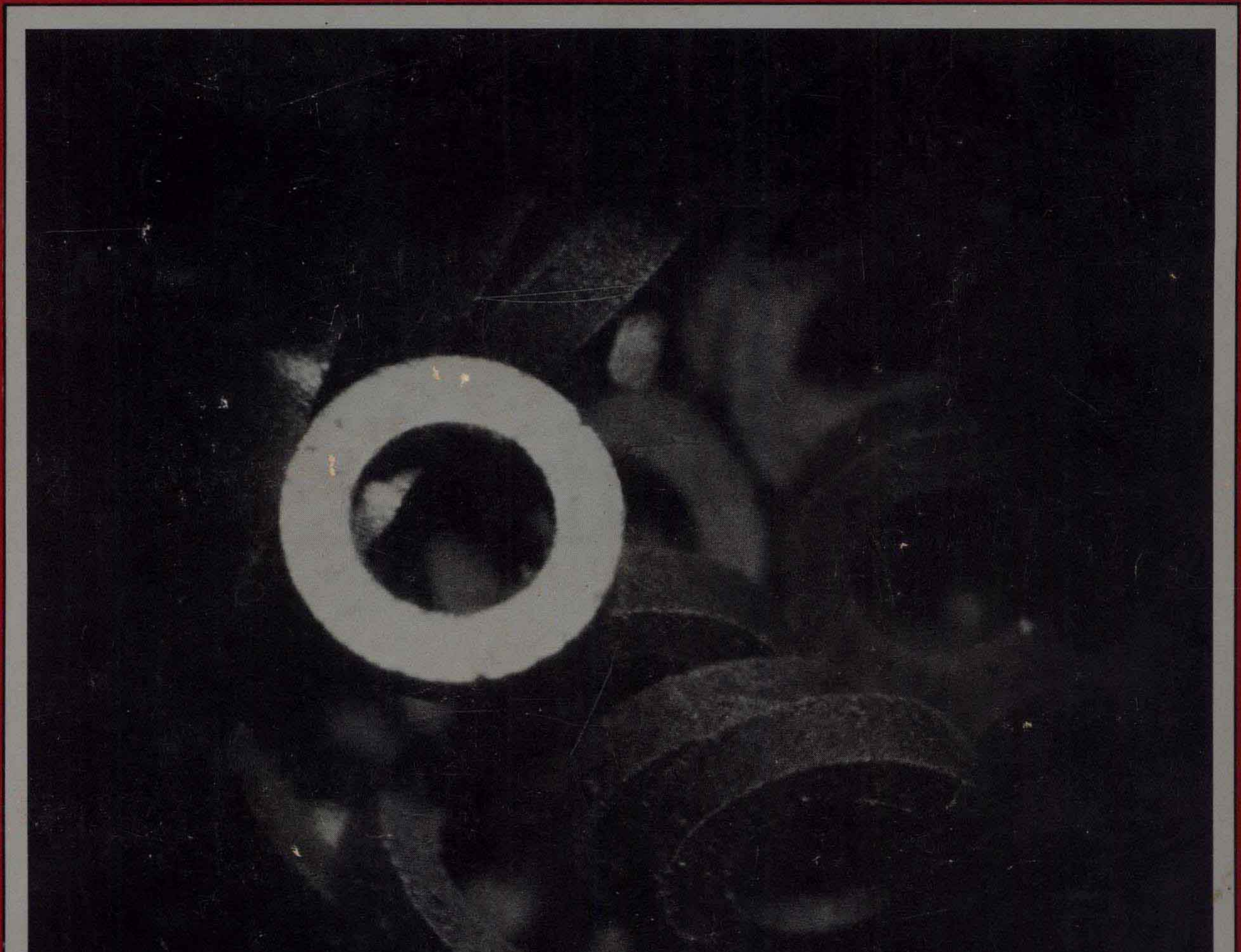
Introduction to BASIC
A Structured Approach

Chris R. Siragusa



Introduction to BASIC
A Structured Approach

Chris R. Siragusa
Cypress College

Prindle, Weber & Schmidt
Boston, Massachusetts

Prindle, Weber & Schmidt is a division of Wadsworth, Inc.
Third printing: November 1980

Library of Congress Cataloging in Publication Data

Siragusa, Chris R
Introduction to BASIC: A Structured Approach

Includes index.
1. Basic (Computer program language) 2. Structured programming. I. Title.
QA76.73.B3S57 001.64'24 79-25401
ISBN 0-87150-289-5

Text and cover design by David Foss, in collaboration with the PWS production staff. Production assistance by Susan London. Text composition in English Times on the Compugraphic Editwriter 7500 by Carol Trowbridge of the PWS production staff. Art drawn by Scientific Illustrators. Cover printed by New England Book Components, Inc. Text printed and bound by Halliday Lithograph.

Cover Photo courtesy Of Electronic Memories And Magnetics Corporation.

Preface

This text was written for students taking their first course in computer science, or their first course in the BASIC language. The only prerequisites for using this text are some high school level algebra and an interest in computer programming. Students from diverse backgrounds and academic disciplines can use this text to learn how to apply programming in BASIC to their studies, whether in business, engineering, natural sciences, mathematics, social sciences, or the humanities. They will also find that the skills they learn in applying computer programming to their fields of intellectual interest will provide them with a solid base from which to undertake further studies in computer science and data processing.

With these goals in mind, this text introduces the BASIC language using the concepts of structured programming and present principles of programming style and documentation. Rather than simply presenting a variety of applications and techniques for writing programs for these applications, this text shows the student how to

1. clearly formulate the logical concepts involved in writing a program (structured programming techniques and concepts);
2. generate readable, easy-to-debug programs (style);
3. document a program.

These concepts are presented in a form of BASIC that is compatible with virtually any type of equipment. However, due to the popularity and usefulness of a number of machines that are capable of using forms of ''extended'' BASIC, the structured statements of IF-THEN-ELSE, WHILE, and UNTIL, plus the logical operators AND, OR, and NOT are treated. These, however, are optional, and are situated in discrete sections of Chapters 6 and 9. Because they are independent, they can also be covered earlier by those who would like to implement them on their systems. The text is written to be machine-independent; any machine-dependent statements are identified as such.

Organization Chapter 1 gives a brief introduction to the history of computer science and an overview of computer systems, indicating the differences between large scale systems, mini-computer systems and micro-computer systems. This chapter is optional, but it is valuable for students unfamiliar with the vocabulary of computer science, and it also provides some background in the development of today's technology.

Chapter 2 introduces structured computer concepts in the way they will be used throughout the text.

Chapter 3 introduces the BASIC language and the concepts of line numbers, instructions, statements, arithmetic, string constants, variable names, and fundamental output.

Chapter 4 introduces intermediate input-output and elementary branching.

Chapter 5 discusses debugging techniques and the importance of good programming style, showing the difference between a well-written modular program and a poorly written unstructured program.

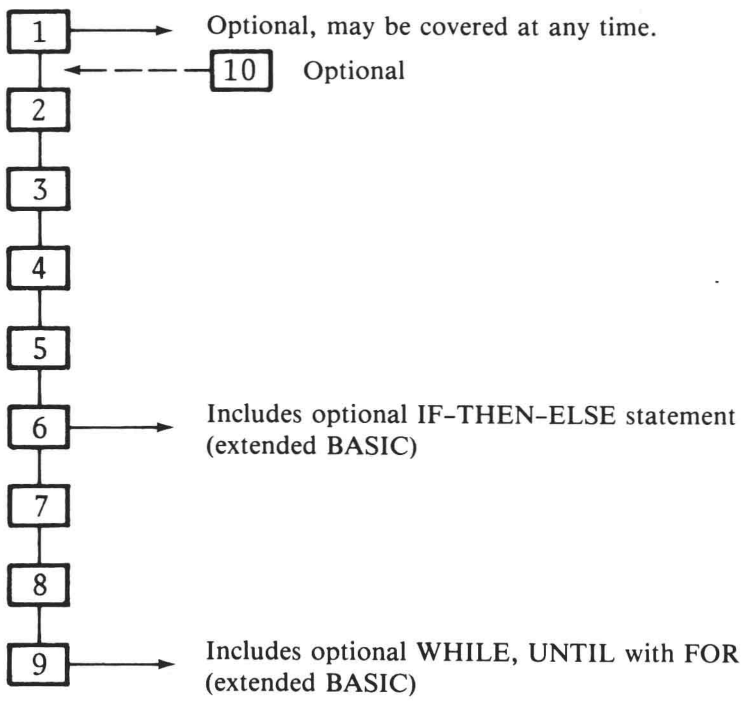
Chapter 6 discusses user-defined and system-defined functions and their use in simulation. Intermediate looping via the FOR-NEXT statement, and the structured IF-THEN-ELSE sequence are introduced.

Chapter 7 introduces subscripted variables, sorting, and plotting.

Chapter 8 introduces matrices, and includes a review of matrix theory.

Chapter 9 discusses subroutines, logical operators, and the structured WHILE, UNTIL, and FOR compounded with WHILE and UNTIL.

Chapter 10 is an introduction to computer science. It requires a firm foundation in mathematics, and is therefore optional, but it gives the student the background to understand the storage of information on different systems, and to be able to read a programmer's reference manual. Should the instructor wish, this chapter can be covered after Chapter 1, as shown in the alternative sequence below.


1
Optional, may be covered at any time.
10
Optional
2
3
4
5
6
Includes optional IF–THEN–ELSE statement (extended BASIC)
7
8
9
Includes optional WHILE, UNTIL with FOR (extended BASIC)

Appendices A and B show standard flowcharting symbols, and BASIC error messages, respectively.

Teaching/Learning Aids Programs and problems have all been class-tested, and are designed for students from various disciplines. Programming exercises are arranged in three categories: business, mathematics and science, and general interest. These exercises are written to be interesting and thought-provoking, and to require the use of the topics most recently discussed in pertinent, applied situations. Within each category, parallel programming logic and BASIC commands are required to complete the programs, so that students and instructors can cover the material in the chapter even if they choose to do only the exercises in one category.

Each chapter features a glossary of terms, and the text contains answers to selected exercises.

A complete solutions manual is available.

Acknowledgements I would like to thank the people whose assistance, talents, and influence are reflected in this text. To Robert Cook (Central Michigan University), Rudy Curd (Appalachian State University), J. Barry DeRoos (Community College of Denver), Bernard Eisenberg (Kingsborough Community College), James Gips (Boston College), Willard Graves (Towson State University), Van Court Hare (University of Massachusetts), Robert Kruse (St. Mary's University, Nova Scotia), Margaret Lewis (Texas Woman's University), Richard Linder (Southwestern College), John Lloyd (Montgomery Community College), John Skelton (University of Minnesota), and Joel Willenbring (Fullerton College), my thanks for their perceptive comments and suggestions. In particular, I am grateful to Chris Liebsack for his expertise, and fortunate to have him as a friend.

My thanks also to Rosemary Butterworth, who typed and proofread the initial and final drafts of the manuscript; to the students of Cypress College, especially Kathy Schow, who tested the exercises and programs in the text; and to Terry Heagney, Theron Shreve, Elizabeth Thomson, David Foss, and everyone at Prindle, Weber & Schmidt for their assistance in producing the text.

Finally, I would like to thank my wife, Karen, for her help in clarifying sections of the text, and proofreading the text, as well as her constant encouragement (and patience) during the writing period; my sons, Eric and Scott, for their understanding —they now hope to eventually write a book of their own; and my parents, Angelo and Ann Siragusa, for their love and understanding and belief in the value of an education.

Table of Contents

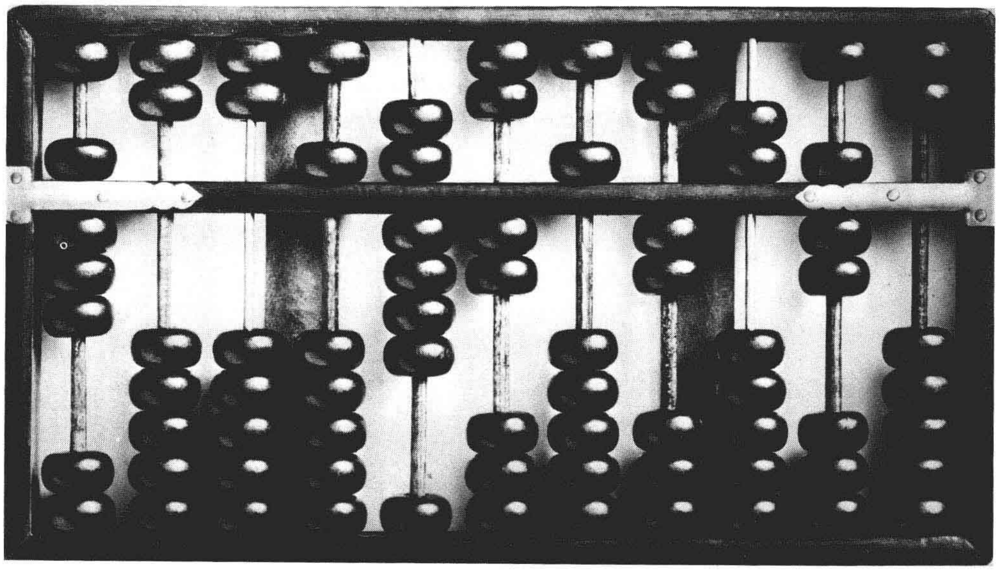
1 Introduction to Computers: Past and Present

1.1 A Brief History of Computers

The modern electronic computer has its earliest origin in the invention of the abacus in approximately 1000 B.C. The abacus, using a system of beads and wires, was the first step in the mechanization of computation.

The abacus is an example of what is called a **digital computer.** A digital computer is a device which stores information in a discrete (or discontinuous) form. On the abacus, numbers are represented by the individual beads. The beads, which are discrete, store the number being represented. This early computer is shown in Figure 1.1.

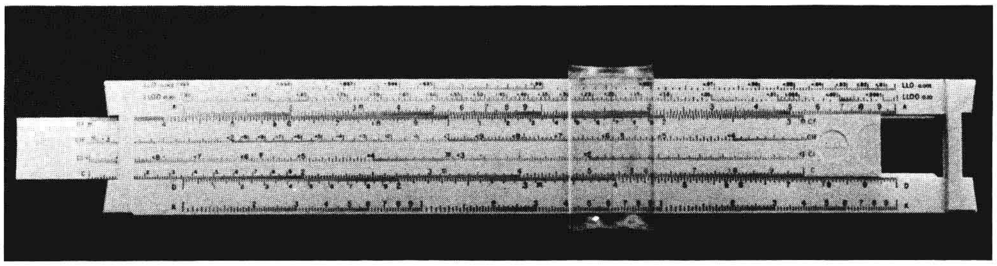
Figure 1.1



The first mechanized computer, the abacus. (Photo courtesy of International Business Machines Corporation)

In 1622, the English mathematician William Oughtred invented the slide rule, which uses sliding scales to perform arithmetic operations. Information is represented in a continuous form on the scales. The slide rule is an example of an **analog computer**. An analog computer stores information in a continuous form. On a slide rule, the slides are continuous and numbers can be found anywhere on the slides, not just at certain points. Figure 1.2 illustrates this early analog computer.

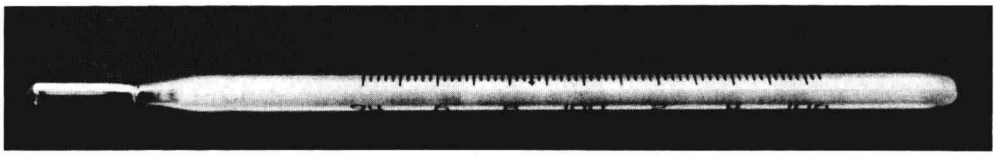
Figure 1.2



The slide rule, an early analog computer. (Photo courtesy of Ronni Linowitz)

Another example of an analog device is an every-day thermometer which has scales showing all possible temperatures within a given range. This analog device is shown in Figure 1.3.

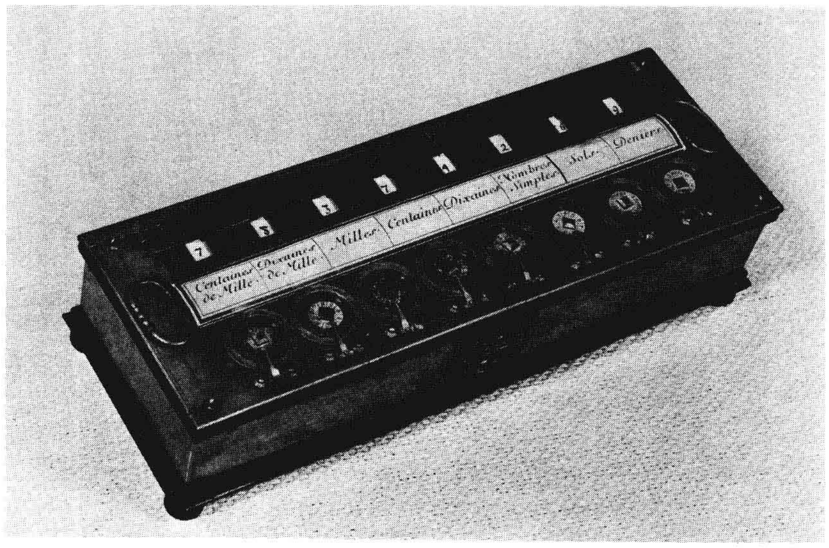
Figure 1.3



The household fever thermometer, another analog computer. (Photo courtesy of Ronni Linowitz)

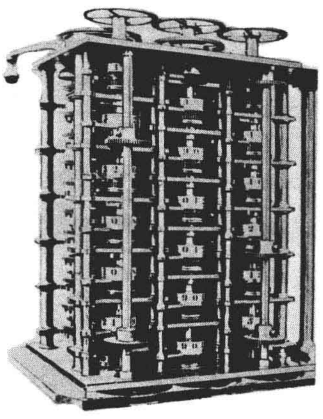
The first mechanical digital computer was designed and built by the French mathematician Blaise Pascal in 1642. Pascal sold about fifty of his mechanical adding machines, which utilized gears, and then went back to what he considered a more practical pursuit—the study of mathematics. His machines were actually the first desk calculators. One is shown in Figure 1.4.

Figure 1.4



The first mechanical digital computer, built by Blaise Pascal. (Photo courtesy of International Business Machines Corporation)

Figure 1.5



Babbage's Difference Engine, designed in 1822. (Photo courtesy of International Business Machines Corporation)

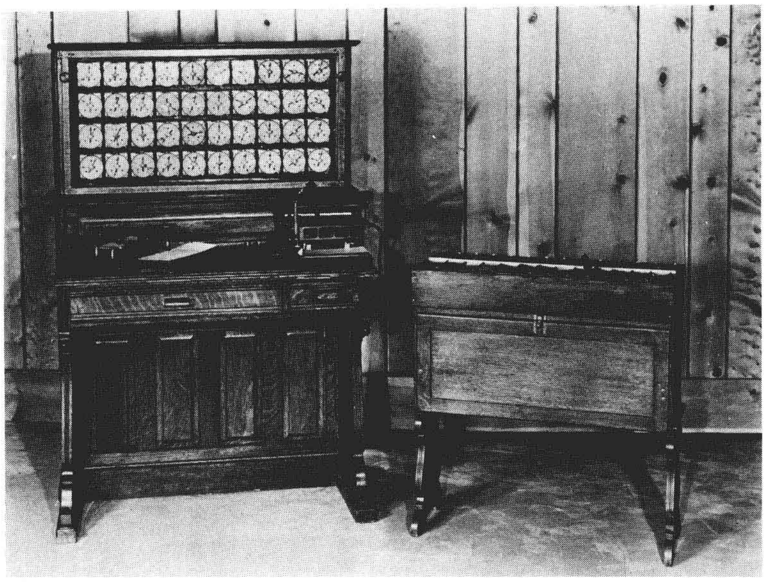
In 1822, an English mathematician named Charles Babbage designed a Difference Engine to handle six-digit numbers and to produce tables automatically (Figure 1.5). It is thought that the accurate computation of logarithmic and trigonometric tables was the primary reason for the construction of this calculating machine. In the early nineteenth century, such tables were computed by hand and were riddled with errors. After a small demonstration of the Difference Engine, the British government promised to subsidize a more powerful model. In 1833, Babbage designed his Analytical Engine, which was the forerunner of today's modern computer. The device was to be driven by steam and use punch cards for the input of data and for instructions (modeled after the Jacquard loom). It was to store the results by means of geared wheels, and then print the results. The project, which was initially financed by the British government, was never completed due to cost overruns, the inability of the technology of the day to produce the machine, and Babbage's inability to stick to a set design. However, Charles Babbage is still considered to be the ''Father of the Computer'' and his ideas are the fundamentals on which today's electronic computers are based.

In 1886, Dr. Herman Hollerith, working for the United States Bureau of the Census, developed a punched card system and a card sorter for accounting. He formed the Tabulating Machine Company. His original card system (shown in Figure 1.6) was not initially used on a computer, but it was eventually used by a company which developed from the Tabulating Machine Company—the International Business Machines Corporation, known world-wide simply as IBM.

During the years 1939-1943, Charles Babbage's ideas became reality with the advent of the Mark I (Automatic Sequence-Controlled Calculator), the beginning of the first generation of electronic computers. It was developed by Howard Aiken and constructed by IBM at a cost of $1,000,000. It was 51 feet long and 8 feet high, and was constructed of electromechanical components. It was obsolete by the time it was completed, because of the rapid advance of technology, but it was important as the first realization of Babbage's concept of a computer.

The first purely electronic computer was built in the years 1943-1946. It was called ENIAC (Electronic Numerical Integrator and Calculator) and was invented by John Mauchly and J. Presper Eckert, Jr. (Figure 1.7). ENIAC had 18,000 vacuum tubes and 1500 relays, plus hundreds of thousands of resistors, inductors, and capacitators, and weighed 30 tons. Each vacuum tube had a volume of 90 cubic inches and used 10 watts of power. ENIAC was hundreds of times faster than Mark I, but

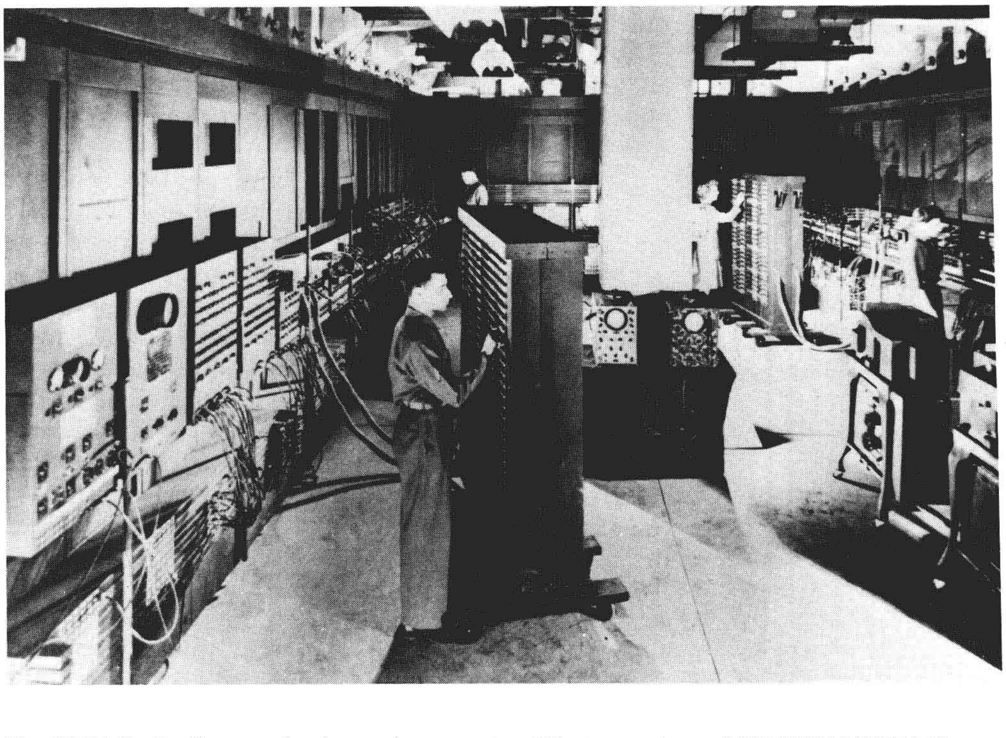
Figure 1.6



Dr. Hollerith's punched card system and card sorter. (Photo courtesy of International Business Machines Corporation)

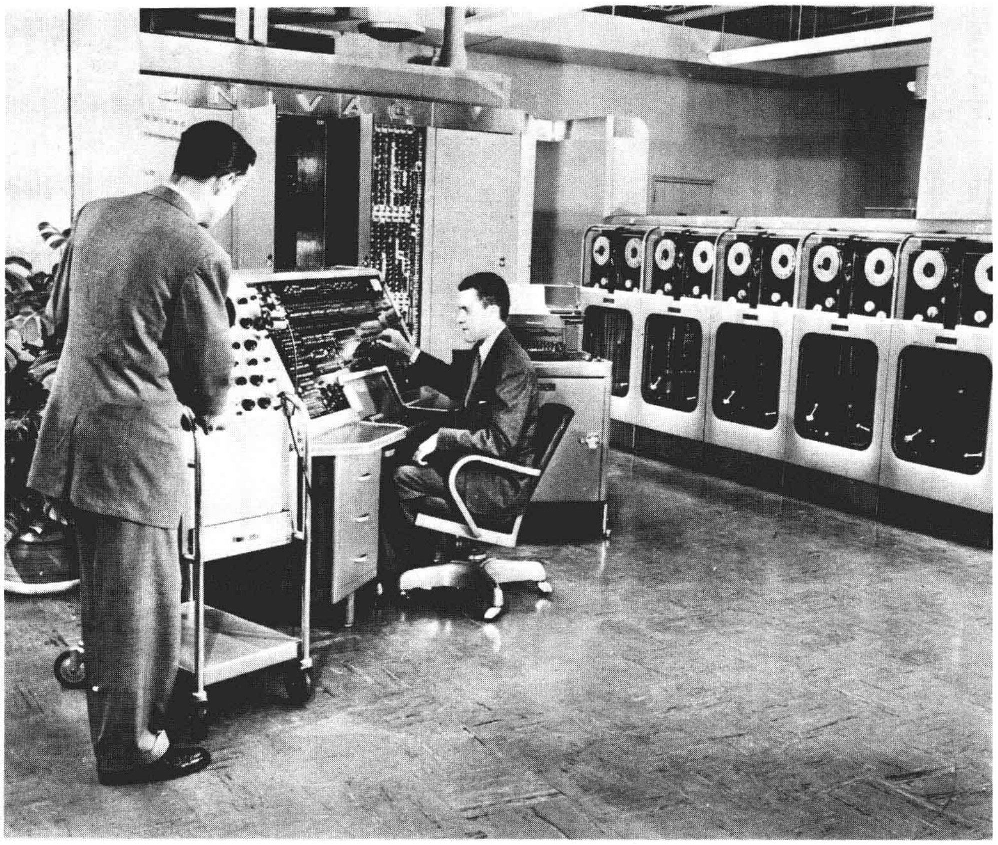
could handle only numerical data. It lacked the facility of storing the entire sequence of instructions needed to handle a complete problem, such as payroll, production scheduling, trajectory simulation, etc., in the memory of the machine. This sequence of instructions for handling a complete problem is called a **computer program**, or simply a **program**.

Figure 1.7



The ENIAC, the first purely electronic computer. (Photo courtesy of SPERRY-UNIVAC)

Figure 1.8

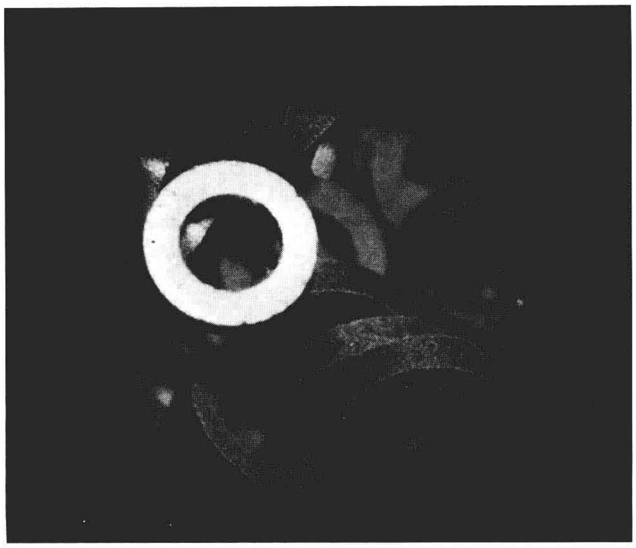


The UNIVAC I, the first commerical computer. (Photo courtesy of SPERRY-UNIVAC)

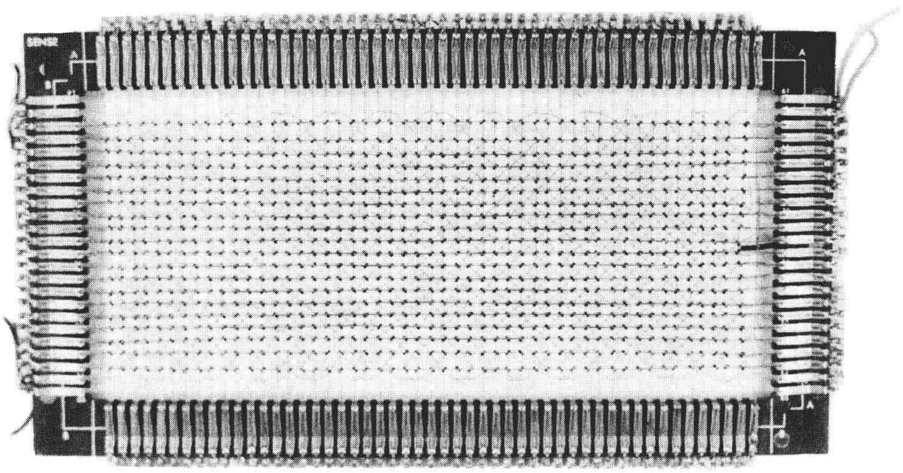
As the technology advanced, the first computers capable of storing entire programs were developed. The first commercial computer, the UNIVAC I (Universal Automatic Computer), was developed by Mauchly and Eckert and was built in 1951 by the Remington-Rand Corporation. It could handle both numerical and alphabetical symbols and employed a mercury storage tank to store the programs. The UNIVAC I (shown in Figure 1.8) was the first large-scale commercial system. Remington-Rand Corporation eventually became Sperry-Rand Corporation and their computers are built by the UNIVAC Division. Computers built in this period, from 1939-1955, are referred to as the **first generation** of computers.

In the late 1950s and early 1960s, the **second generation** of computers appeared. These machines used transistors, core memories, magnetic tapes, and magnetic drums (these concepts will be covered in more detail in Section 1.2). Transistors took the place of vacuum tubes. Individual transistors were no longer wired together, but all of the components were put on a single piece of silicon, called a **chip**. This integration of many components on a single chip is called **Integrated Circuit (IC) technology**. The number of electrical components which could be included on a single chip doubled every year until about 40 components were integrated on a single chip. For this reason, the late 1950s and early 1960s are called the period of Small Scale Integration (SSI). During this time, IBM and UNIVAC produced several large-scale computer systems capable of performing two million additions per second. However, computer systems still cost upward of five million dollars and were not economically feasible for small- and medium-scale users. Figure 1.9 shows a second generation core memory and a second generation core plane.

Figure 1.9 (a)



(b)

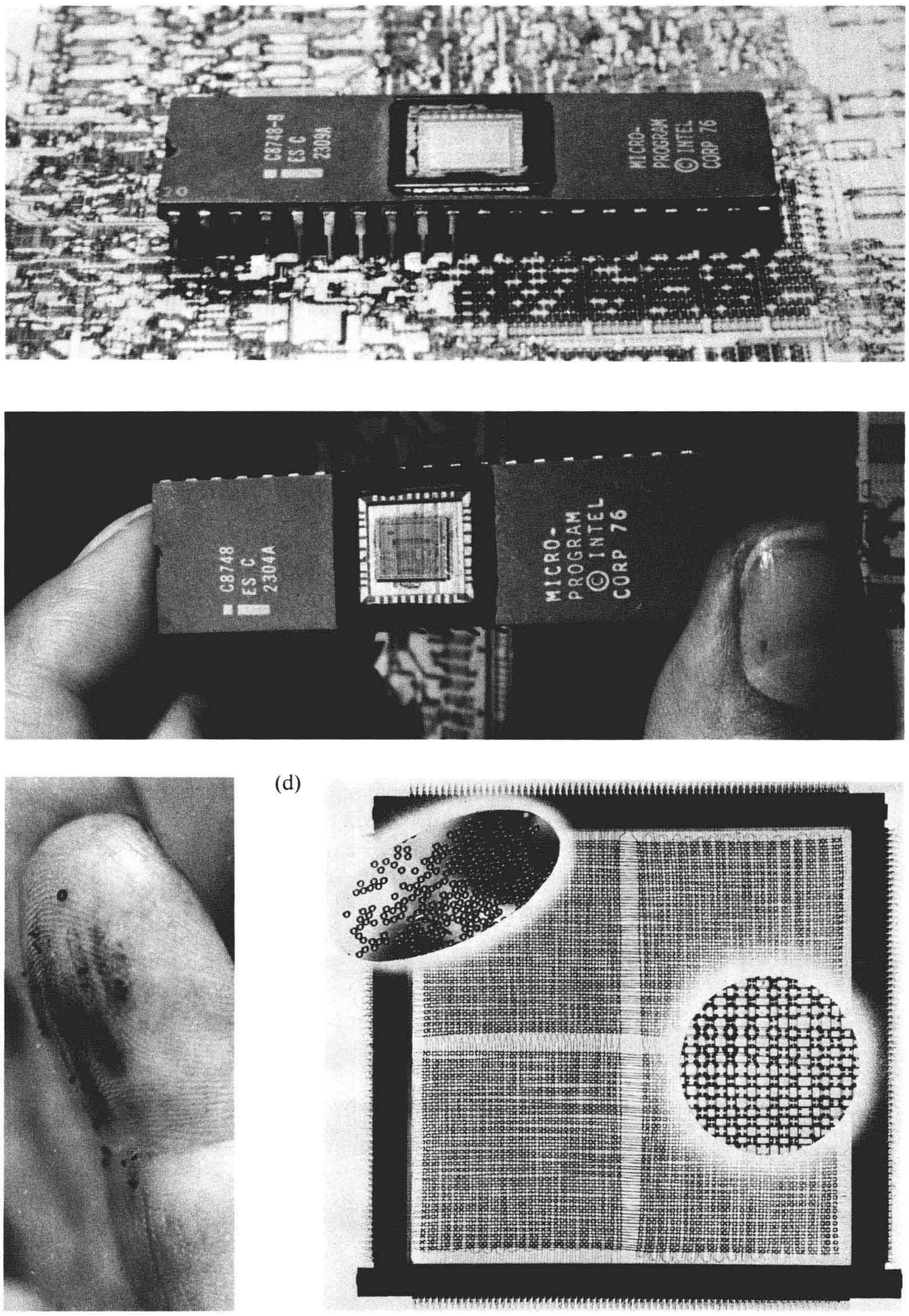


(a) Second-generation core memory (photo courtesy of Electronic Memories and Magnetics Corporation); (b) A second-generation core plane (photo courtesy of Professor Robert L. Azen, Cypress College).

In the middle 1960s and early 1970s the **third generation** of computers evolved. This generation came in two stages. The first stage was that of Medium Scale Integration (MSI) in which the number of components per chip increased to several hundred. The second stage was that of Large Scale Integration (LSI) in which chips able to hold 20,000 components in a space of less than one square inch were developed. Smaller core memories and new memories (thin film) were developed. Figure 1.10 shows a chip which can hold 4000 computer words, and a third generation core memory. In this second stage, companies like IBM, UNIVAC, Control Data Corporation (CDC), Digital Equipment Corporation (DEC), Hewlett-Packard, National Cash Register (NCR), and a host of others, developed not only large-scale systems, but also medium-scale systems, called minicomputers, for the medium-scale business and industrial user. In 1965, Digital Equipment Corporation introduced a minicomputer, the PDP-8 (Figure 1.11), with a price of about $20,000. As the components became miniaturized, the cost of the systems came down.

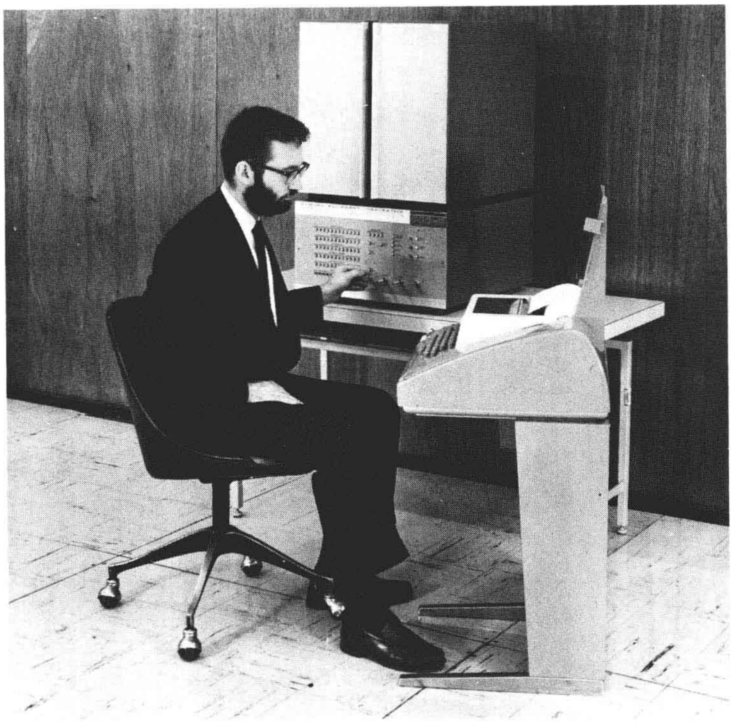
Figure 1.10

(a) (b) (c) (d)



(a) The 2304A, 4000-word microchip by INTEL Corporation. The chip is the small square mounted on the black holder. The holder is resting on a photograph which shows a 500 × magnification of the chip. (Photo courtesy of INTEL Corportation); (b) the size of the 2304A chip relative to the human hand. (Photo courtesy of INTEL Corporation); (c) third-generation cores, compared to a human finger. (Photo courtesy of Electronic Memories and Magnetics Corporation); (d) third-generation core plane. (Photo courtesy of International Business Machines Corporation).

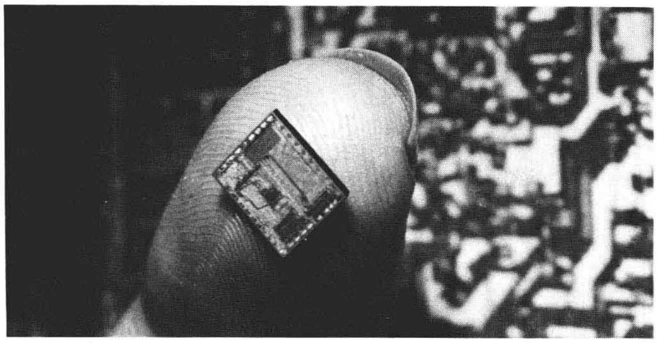
Figure 1.11



The original PDP-8 minicomputer. (Photo courtesy of Digital Equipment Corporation)

Fourth generation computers, which began appearing in the late 1970s, are characterized by chips which can hold up to 100,000 components. Twenty-five hundred microminiature transistors can be placed on a chip 1/6″ × 1/8″, where each transistor is equivalent to one of the old vacuum tubes. New memories (bubble) capable of holding one billion pieces of information on a chip one inch square are used. Complete computer systems are being built on one or two chips. This miniaturization has led to the development of smaller systems, called microcomputers. Microcomputer systems presently cost from $450 to $2000, and are thus affordable by individuals and small businesses. Figure 1.12 shows a microcomputer chip built by INTEL Corporation which is used in many microcomputers.

Figure 1.12



A microcomputer chip made by INTEL Corporation, which is used by many microcomputer manufacturers. The chip contains all of the basic components of a computer system, and is as powerful as the UNIVAC I shown in Figure 1.8. (Photo courtesy of INTEL Corporation)