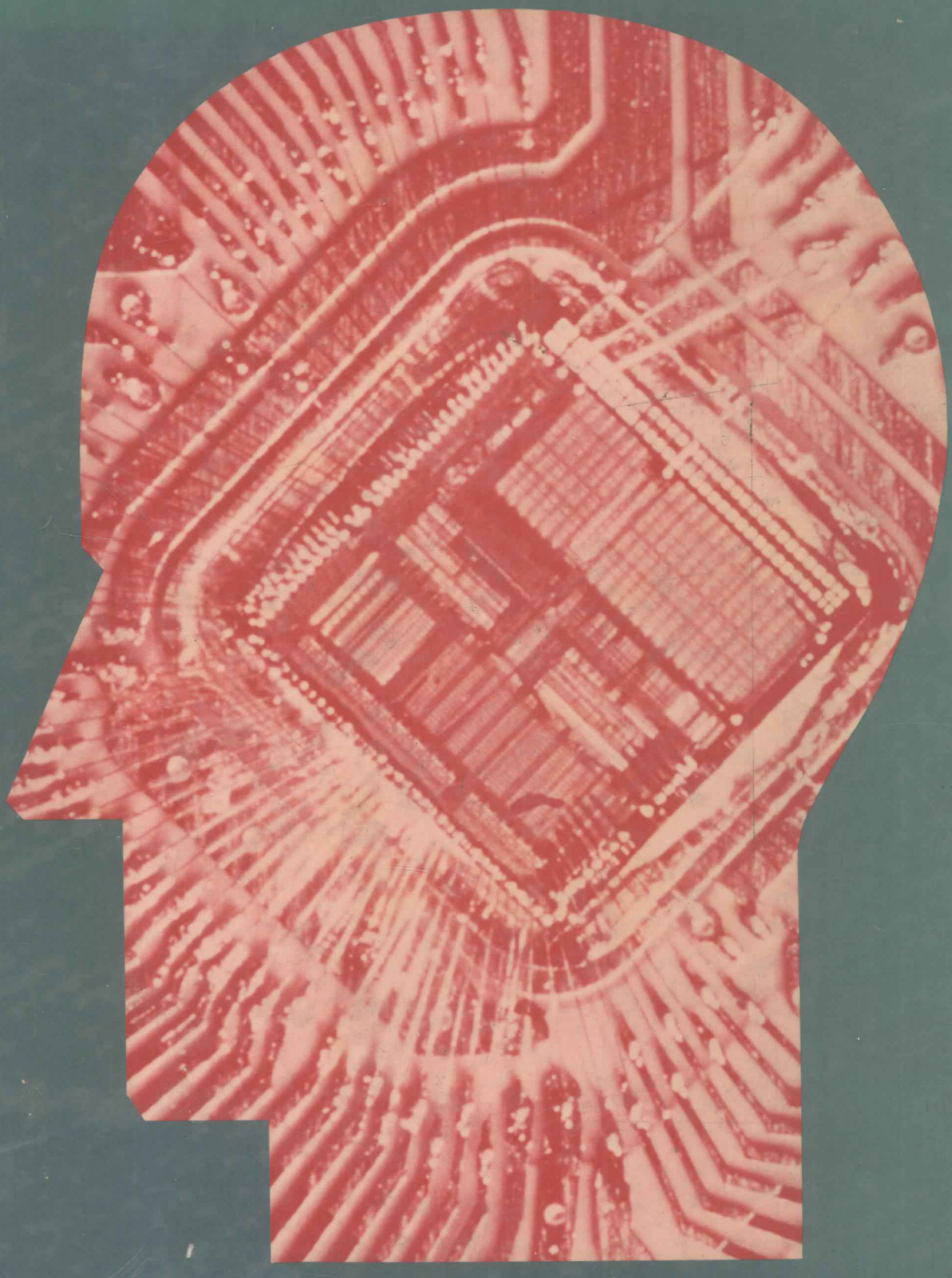
The Intelligent Microcomputer

Roy W. Goody

Second Edition



ROY W. GOODY
Mission College, Santa Clara, California

THE INTELLIGENT MICROCOMPUTER

Second Edition

SRA®
SCIENCE RESEARCH ASSOCIATES, INC.
Chicago, Henley-on-Thames, Sydney, Toronto
An IBM Company

Acquisition Editor	Michael Carrigg
Project Editor	Byron Riggan
Compositor	Graphic Typesetting Service
Illustrator	John Foster
Cover Designer	Harry Voight
Text Designer	Judith Olson

Cover Photo courtesy of Hewlett-Packard Company

Appendixes I–VI (pp. 413–429) reproduced by permission of Intel Corporation

Library of Congress Cataloging-in-Publication Data

Goody, Roy W.
The intelligent microcomputer.

Includes index.
1. Microcomputers. I. Title.
QA76.5.G627 1986 004.1'6 85-14371
ISBN 0-574-21615-4

Printed in the United States of America

10 9 8 7 6 5 4 3 2

To George and June Goody

preface

If learning about the complex field of microcomputers can be compared to scaling a high peak, then this book is written as a staircase. It is designed to provide the beginning- to intermediate-level technician or electronics-technology (ET) engineering student a gradual, step-by-step, upward learning path, eliminating the sharp rises and gaps that often impede progress. The Contents lists the steps in the staircase and divides the material into 27 easily scaled chapters. Depending on the pace of presentation, these chapters might be divided appropriately into two sections for a two-semester or two-quarter series of courses.

Emphasizing the major aspects of hardware design (configuring), program development, interfacing, and applications, the material in this book is based on the *inventor's approach*. That is, on successful completion of the material, you will be able to design and troubleshoot a simple microprocessor-based system starting completely from scratch. The scope and depth are sufficient for both the technician, who must test and troubleshoot the system, and the engineer, who must design and develop the system.

Little or no previous knowledge of computers is required. Prerequisites are a basic background in ac/dc and semiconductor electronics, experience in the use of the hexadecimal number system, and a working knowledge of digital circuits and gates. No review of basic digital concepts and number systems is included; it is assumed that you have completed a course in basic digital concepts before undertaking microcomputer studies.

This book is based on the popular Intel family of microprocessors (the 8080, 8085, 8048, 8086, and 80286 to name the most prominent). However, the bulk of the presentation will center on the popular 8-bit 8080/8085, microprocessors which represent the ideal level of complexity for those just entering the field. The study of these real devices and real systems, whose characteristics can be reproduced in the laboratory, is deemed to be the most effective way to focus the mind on this complex subject. The basic principles gained from a study of the Intel product family can, of course, be easily applied to any comparable microprocessor on the market. (*The Versatile Microcomputer* is a Motorola 6800-based version of this text.)

To those who are familiar with the first edition, this second edition has been updated to reflect the latest technology, and it has been expanded in both scope and depth. To give several examples, there are sections on CAD/CAE, operating systems, integrated RAMs, layered networks, structured programming, and in-circuit emulation (ICE). The number of worked-out examples and sample programs has been greatly expanded, and there is a new chapter covering the 80186 and 80286 microprocessors.

A major addition is the inclusion in Chapters 14 through 21 of a number of *machine assembly updates*, providing information to keep pace with the increasing availability of machine assemblers. Although all machine assembly was done on an Intel Series II Development System to demonstrate the tools and methods used within industry, the machine assembly process is very similar to those presently available for the more popular personal computers (such as the IBM PC and Apple II).

To enhance the software development skills of the student, the Questions and Problems sections at the end of each chapter suggest nearly 60 programs (in addition to those presented in the text) that can be written and tested. The solutions to all suggested programs are given in the *Instructor's Guide*. All programs listed in this text and in the *Instructor's Guide* can be run without modification on the Intel SDK-85 single-board computer (with only minor modifications required to run on any 8080/8085-based single-board computer).

To summarize, the *content* of this book is grounded in fundamentals: design, troubleshooting, and interfacing in a balanced hardware/software environment. The *theme* of this book, however, is quite different, for it keeps an eye on the future, and in the future one subject will dominate the field of computers: *artificial intelligence*.

Therefore, to further increase enthusiasm and to make the material more readable, this book takes full advantage of the notion of the computer as an intelligent machine (an android). Indeed, if you set out to build a functioning intelligent machine, what major steps would you follow? First you would fashion the anatomy of the system (assemble the hardware). Next you would study its basic bodily processes and metabolism (basic processing action). You would then

bring it to life (initiate processor action), send it off to school to be taught (programmed) many useful skills, and finally train it to live successfully in the real world (interfacing and applications). These are also the steps in the staircase along which this book is organized.

Also within the theme of artificial intelligence (AI), the text makes occasional reference to the evolutionary parallel between human and computer. The author has chosen this parallel as a natural way to introduce a number of computer concepts. No attempt is made to convince you that a machine will someday be the equal of a human being. The subjects of artificial intelligence and natural selection are used primarily as vehicles to motivate your interest and perhaps to make the material more exciting and fascinating. On the other hand, the similarities between computers and people should in no way be construed as mere fantasy; moreover, it is predicted that after completing this book you will come to regard your home computer a little more like your family pet (a living creature) and a little less like your family car (a simple tool).

Roy W. Goody

THE INTELLIGENT MICROCOMPUTER

Second Edition

contents

chapter 1

The Microcomputer: An Overview

The microcomputer has flourished because—like its animal-kingdom counterpart—it found a fertile niche in the electronic environment and survived by being the fittest of the species.

In this chapter we will look at the historical development of the microcomputer and examine its fundamental nature. As you will see, we can learn a great deal about the computer by studying ourselves.

THE MICROCOMPUTER REVOLUTION

By the turn of this century the first industrial revolution, which began in England around 1720, was well under way, and machine power was replacing muscle power at an ever-increasing pace. Since the early 1970s a second and far more profound industrial revolution has gathered strength. This time, however, machine power will enhance and replace not the muscle power of the human species but the *brain* power.

The machine we are talking about is, of course, the computer. But computers are not all that new. Why then has the microprocessor—essentially a computer on a chip—ushered in the second industrial revolution?

Strangely enough, the answer to this question can be found on the Salisbury Plain in southwest England—the site of Stonehenge. This curious arrangement of 30-ton stones, each hewn from a distant quarry and transported hundreds of miles, is actually an ancient neolithic computer, constructed 500 years after the Egyptian pyramids to predict eclipses and other celestial events. Clearly, if today's computers were of Stonehenge design—requiring 30-ton components and hundreds of years to design and construct, fashioned from hard-to-get materials, dedicated to a single purpose, slow, inaccurate, and very limited in power—the second industrial revolution would not be taking place. But today's microprocessor-based computers are just the opposite. They are inexpensive, ultrasmall, lightweight, multipurpose, highly accurate, breathtakingly fast, and incredibly powerful—and the second industrial revolution is under way.

MICROPROCESSORS AND EXISTING SYSTEMS

To be more specific, microcomputers are successfully filling a wide gap in the electronic-design spectrum between ordinary arrays of gates and registers (called *combinational* or *random logic*) and minicomputers. Toward one end of the gap, microcomputers are replacing hardware with software, and toward the other end, they are replacing numerous discrete computer components with a handful of very large scale integrated (VLSI) chips, each containing 50,000 or more transistors on a single chip of silicon. Any combinational-logic circuit of 30 or more gates is a prime candidate to be replaced by a microprocessor, and any computer system of today—including mainframe computers—may soon be replaced by a handful of VLSI blocks.

MICROPROCESSOR DESIGN

Microprocessors are unique in the world of electronic design because successful designers and users must have a balanced knowledge of hardware, programming, and interfacing. No longer do we have the luxury of specializing either in hardware design or pure programming. Sometimes a design problem is best solved with extensive programming, using only the simplest of external hardware. At other times a careful combination of VLSI blocks will provide the best overall solution. Once this balanced approach is accepted, the rewards will be faster, more powerful, and less costly system designs.

MICROPROCESSORS AND MICROCOMPUTERS DEFINED

In a single sentence, a *microcomputer* is a system containing a *microprocessor.* A *microprocessor* is a VLSI programmable logic device on a single silicon chip, less than $\frac{1}{4}$ inch on a side, usually containing all necessary computer com-

ponents except memory and input/output (I/O) ports. A *microcomputer* is an entire computer system, including a microprocessor, external memory, and I/O devices. Occasionally, the entire computer system is integrated on a single chip (the 8048 family), although means are usually provided for adding extra memory and I/O ports.

Compared to mini- and mainframe computers, a microcomputer is usually smaller and less expensive, and often does not include the wide variety of expensive peripherals, such as cathode-ray tube (CRT) or disk memory. Microcomputers normally require less memory, are slower, and often are dedicated to a specific task, whereas mini- and mainframe computers are usually very high speed, general-purpose systems. However, the realm of the microcomputer overlaps into the area now dominated by the minicomputer, and soon will push into the area occupied by mainframe systems. Clearly it is difficult to define a system that is constantly changing; and, compared to human beings, computers are changing and adapting at an explosive pace!

MICROCOMPUTER APPLICATIONS

Even more important than the replacement of existing circuits and systems are the thousands of applications lying within the design gap between combinational-logic arrays and minicomputers—applications only the microprocessor can bring to life. The applications are so vast that the microprocessor revolution has already spawned an electronic age, in which an army of willing servants watches over us from morning to night. We wake up to a microprocessor-controlled alarm clock, read a newspaper that was edited and printed by a microcomputer-based word processor, and watch the morning news on television as a microprocessor fine-tunes the picture. We leave our homes guarded by a microprocessor "watchman," drive to work as a microprocessor instantly adjusts the car's timing and fuel/air mixture for optimum performance, and converse over a microprocessor-controlled CB radio. Our commute is speeded up by microprocessor-based traffic-control systems, our on-the-job productivity is increased by computerized inventory systems, and our workrooms are environmentally controlled by a microprocessor. We shop at a store where an intelligent (microprocessor-controlled) cash register inputs data to a central computer system that automatically inventories and orders merchandise. We go out for dinner and have our meal ordered and cooked under microprocessor control, and our drinks dispensed with a microcomputerized mixing machine. Then we go to bed and dream.

When we wake up it will be the future and our alarm will talk to us in perfect English. Our home computer will accurately forecast the weather and diagnose our ailments. We will realize how much we have come to rely on our computer for its *judgment* as well as its knowledge, and it will be hard to tell whether we are still dreaming.

A BRIEF HISTORY

Like most inventions, the microprocessor resulted from the gradual blend of many scientific trends. Those most important to the development of the microprocessor were the mathematical, electronic, and computer trends. As shown in Figure 1.1, several important milestones finally led to today's advanced microprocessor.

- *The early days:* The first calculations were done on the human hand. From this simple beginning the familiar decimal, binary, and hexadecimal number systems eventually evolved. The first mechanical device to make use of number systems was the *abacus,* a calculating device that dates back before the birth of Christ and is still used today.
- *1642—Calculating machine:* Blaise Pascal invented the first "desk calculator." It was strictly a mechanical device, using systems of gears to add and subtract. Since the precision machining of parts was still many years away, the idea slowly died. (But Pascal's name did not die away, for a popular high-level computer language is named after him.)
- *1801—Automatic loom:* Joseph Jacquard's idea revolutionized the weaving industry and was destined to resurface many years later in the computer industry. It was an automatic loom that used punched cards (IBM cards!) to control the pattern.
- *1833—"Analytical engine":* Charles Babbage, more than any other computer pioneer, deserves the title "father of modern digital computers." His "analytical engine," developed to calculate and print mathematical tables, incorporated many of the principles of modern digital computers. Babbage was the first to envision the stored-program concept, in which all numbers *and* instructions were read before calculations began. In other words, once programmed, the machine worked without human intervention. Unfortunately, for a number of practical reasons that plague all inventions ahead of their time, it was never developed beyond the prototype stage.
- *1854—Boolean algebra:* Can formal logic be described mathematically? George Boole discovered that it could, and he developed a symbolic form of logic called *Boolean algebra,* a subject familiar to every student of digital electronics. The door to computer design was now wide open.
- *1890—Electric tabulating machine:* Herman Hollerith developed the first true data-processing machine. Using

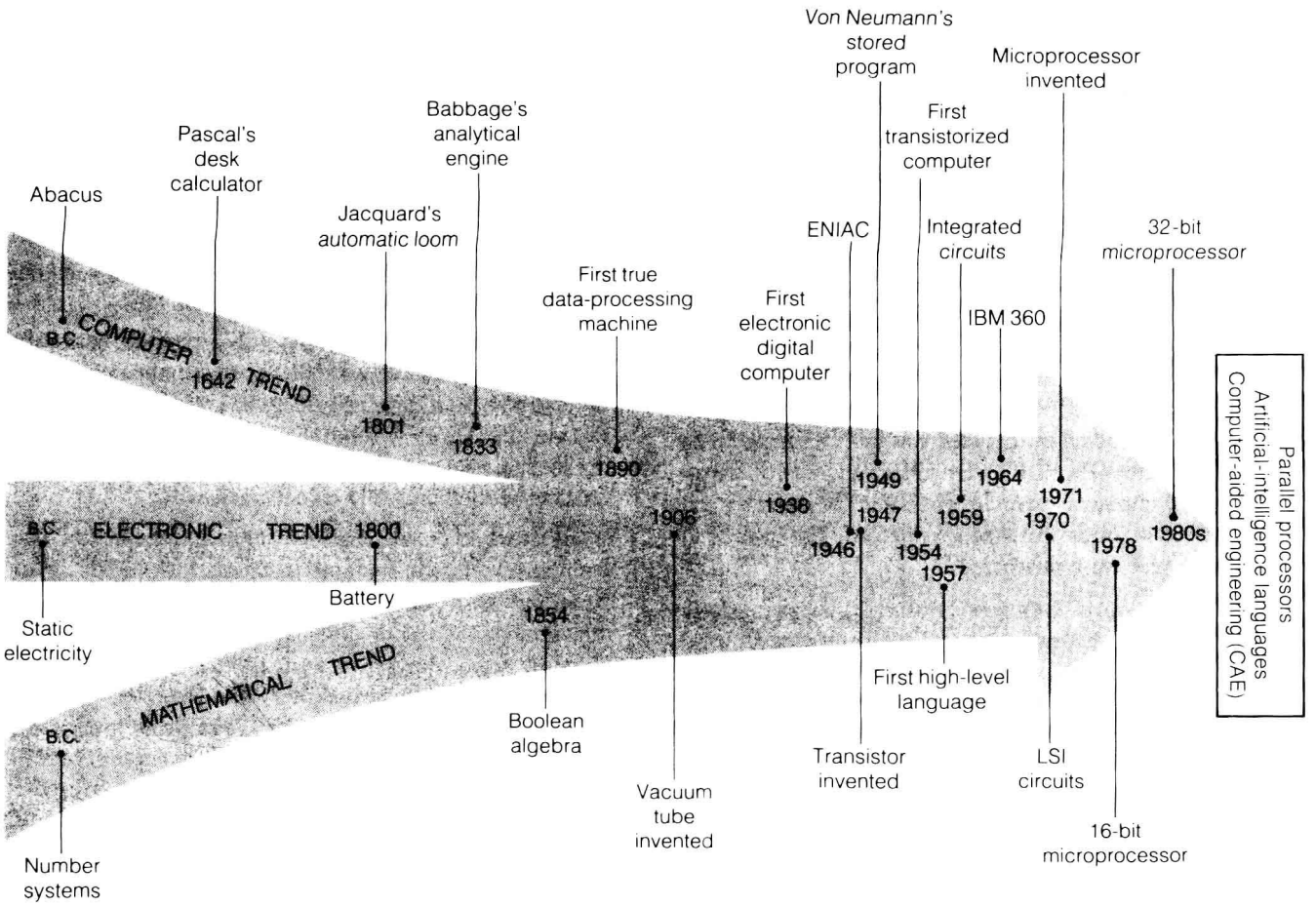


Figure 1.1 Technological trends and milestones in microprocessor development.

his machine, the task of tabulating the 1890 census was reduced from 20 to 3 years.

- *1906—Vacuum tube:* The electronic pathway began in earnest with the application of the triode vacuum tube, invented by Lee De Forest in 1906. Mathematical manipulations could now be done electronically rather than mechanically, with a quantum jump from seconds to milliseconds in processing speed.
- *1938—Electronic digital computer:* John V. Atanasoff formulated the basic ideas for computer memory and associated logic, and built the first electronic digital computer. Based on vacuum tubes, it paved the way for all work to follow.
- *1946—Large-scale electronic digital computer:* Prompted by the wartime need to calculate ballistic tables to produce trajectories for artillery and bombing, the U.S. Army funded the Electronic Numerical Integrator and Calculator (ENIAC) project. Completed in 1946, ENIAC was the first large-scale electronic digital calculating machine. By today's standards it was a monster. Composed of 18,000 vacuum tubes, it weighed in at 30 tons, occupied 1,500 square feet, and consumed 130,000 watts of power. However, it could multiply two numbers in about 3 milliseconds, a thousand times faster than ever before—and without the use of a single moving part. It was turned off for the last time in 1955.
- *1949—Stored-program computer:* Although ENIAC could perform individual mathematical operations at high speed, it had to wait for each instruction to be entered by its human operators. Intent on removing this human factor, John von Neumann picked up on the stored-program concept first conceived by Babbage and proposed placing computer instructions as well as data in the computer's memory. Whenever a sequence of instructions was to be performed, the computer could read in each instruction from memory without waiting for human intervention. Storing the program inside the computer along with the data allows today's computers to operate at high speed (and distinguishes a computer from a calculator).

With the stored-program concept, the last major hurdle to modern computer design was crossed, and in May of 1949 the first digital computer based on the stored-program concept went into operation. Named the Electronic Delay Storage Automatic Calculator (EDSAC), it set the stage for all computers to follow.

- *1954—Transistorized computer:* Although based on electronics, ENIAC and EDSAC were still of "Stonehenge" design—far too big, bulky, and power consuming to command widespread attention. In 1947, however, a breakthrough was made that can only be described in fairy-tale terms, for like Alice in Wonderland, the solid-state research it set in motion was destined to shrink the size of computers a thousandfold and more. Invented by John Bardeen, W. H. Brattain, and W. B. Shockley at Bell Laboratories in 1947, the transistor was the seed from which the second industrial revolution sprouted.

  The first product of this seed sprouted in 1954 with the introduction of the TRAnsistor DIgital Computer (TRADIC). By 1960 hundreds of transistorized computers were in operation, processing data faster and at lower cost than ever before. The days of the vacuum tube were numbered.

- *1957—High-level language:* Primarily because of their awesome size, the early vacuum-tube computers quickly acquired a public image of "giant brains," fearsome machines to be viewed with apprehension and mistrust. This image was largely undeserved, of course, for these early computers were very crude, and in one area in particular—languages—they were downright primitive. The only language these early machines understood was machine language—the language of ones and zeros—a language that made programming a cumbersome, error-prone, and difficult endeavor.

  The first major breakthrough in language development was made by an IBM research team headed by John Backus. Primarily interested in developing a language to solve mathematical calculations, the team devised a way of writing a program using mathematical notation instead of machine language. Using common typewriter symbols to write and enter the program, the computer would then translate (compile) the sequence of symbols into the required machine-language instructions.

  Introduced in 1957, the language was called *FORTRAN* (for FORmula TRANslation) and is still in widespread use today. By 1960 numerous high-level languages were in use, including COBOL (COmmon Business-Oriented Language), which gave the business community many of the same advantages that FORTRAN gave the scientific community.

- *1959—Integrated circuit:* By applying the principles of photolithography to flat surfaces of silicon, and by developing the method of solid-state diffusion for introducing the impurities that create *p* and *n* regions, engineers found they could construct entire circuits, consisting of many transistors, on a single chip of silicon. This was the basis of the integrated circuit, a technology that was to show the same explosive growth in sophistication as took place in the human brain during the end of the last ice age.

- *1964—Integrated-circuit computer:* On April 7, 1964, IBM introduced the standard mainframe computer, the System/360 (called 360 because the system was said to encompass the full range of scientific and business applications). Using highly reliable, mass-produced, integrated circuits, it could perform in 1 second nearly a half million computations at a cost of less than 10 cents. The System/360 also introduced a number of new input/output and auxiliary storage devices.

- *1970—Large-scale integrated (LSI) circuit:* From the early 1960s to the early 1970s, the maximum complexity of integrated circuits doubled approximately every 18 months. By 1970 more than 15,000 transistors could be etched onto a single chip of silicon, an achievement that made the handheld calculator feasible.

- *1971 to present—era of the microprocessor:* In 1971 the computer, electronic, and mathematical trends came together in a unique way, and the microprocessor emerged on the scene—initially with little fanfare. In less than 15 years, though, it went from a simple 4-bit LSI device developed for calculators (the Intel 4004), to the iAPX386, a 32-bit SLSI (super large scale integrated) enhanced microprocessor family that forms the heart of a system resembling a mainframe computer.

- *Sometime in the future—first true intelligent machine:* In the field of microprocessors, the future blends with the past so quickly that no historical survey would be complete without a word about what lies ahead. Most far-reaching of all is a fourth scientific trend that is now merging with the ongoing development of the microprocessor. This fourth trend, known as *artificial intelligence,* will combine with the new parallel array processors to produce the true intelligent machine. Unlike the human brain, which must depend on the relatively slow process of biological change, the intelligent machine made of silicon is under no such restrictions. We can only wonder where the new technology will lead us. Soon a true learning machine will emerge, *one able to modify its program based on learning experience*. What would be the result if we taught two such learning machines to play chess, and what if they were pitted against each other, playing games at the speed of light for a month or a year? At the end of the time, what would we find? Perhaps a new way of thinking, or a new philosophy, or a new mathematics—or perhaps something we would not be able to understand at all!

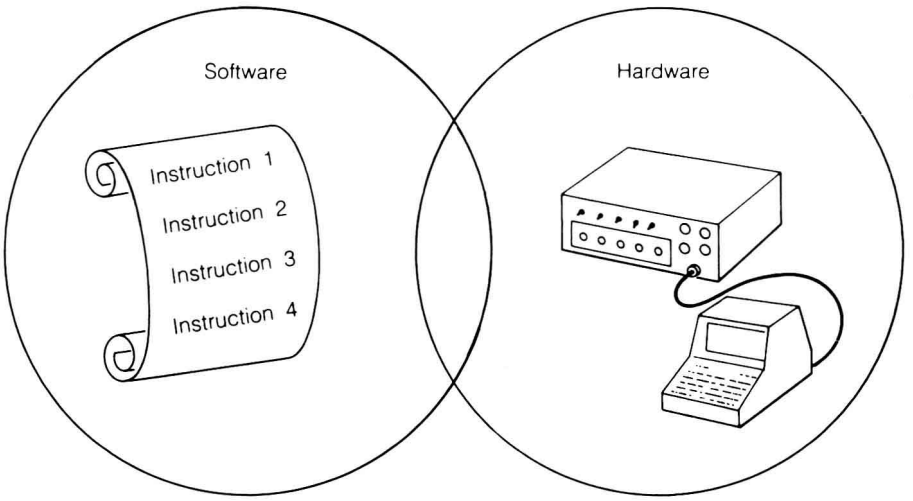


Figure 1.2 A computerized solution to a problem—both software and hardware required.

THE FUNDAMENTAL PRINCIPLES OF COMPUTER ACTION

The apparent similarity between people and computers presents us with an exciting possibility: If computers and humans do things in a similar way, then to develop the more basic concepts of computer action perhaps we can begin by studying ourselves.

What separates us and other members of the animal kingdom from the world of inanimate objects? The answer is very straightforward: we perform tasks; we do things.

To use an everyday example, consider a major-league outfielder catching a baseball. Even the most cursory analysis of this simple task reveals that it is composed of individual steps, taken in sequential order:

Track ball
Run under ball
Raise glove
Catch ball

Each step in the sequence commands a specific operation. In computer terminology, each command is called an *instruction*. To complete the task of catching a baseball, then, we simply go through the instructions in order. *A computer performs a task in precisely the same way.* Therefore, by studying our own actions, we already have uncovered perhaps the most fundamental of all computer concepts. Seven of these concepts are described below.

1. *A computer performs a task by processing a sequential list of instructions.*

Of course, it is important for us to write the list of instructions in the proper order. If any of the steps is incorrectly listed, the task probably could not be completed.

2. *To carry out a task by way of computer action, we require both the list of instructions (the software) and the physical circuitry (the hardware).*

To catch a baseball, we require two major items: the list of instructions and the collection of physical components (player, baseball, and glove). When applied to computers, the sequential list of instructions is known as *software;* the physical computer circuitry and peripheral components are known as *hardware* (see Figure 1.2).

3. *Each instruction given to a computer generally consists of a verb portion (the operation code) and an object portion (the operand).*

As listed below, each instruction written for a person is also made up of two parts—a verb or action portion and a noun or object portion:

Verb	*Object*
track	ball
run under	ball
raise	glove
catch	ball

4. *As each instruction is carried out (executed), the operation code (verb) directs the activities of the operand (object).*

As each instruction of our baseball routine is processed, the verb or action portion directs the activities of the object portion, and the instruction is carried out—or *executed.* Since this is true of all instructions carried out by people, it is also true of all instructions carried out by computers.

5. *Programming a computer means entering the proper sequence of instructions into its memory.*

During training, a baseball player quickly commits to memory the sequence of instructions for catching a baseball. In other words, the player has *learned* the sequence of steps. In computer terminology, learning is known as *programming*, and it consists of storing the sequence of instructions in memory. As previously noted, the concept of a stored program—that is, placing the instructions in memory before they are needed—was one of the great historical advances made in computer technology.

6. *A computer consists of five basic hardware blocks: input, output, memory, arithmetic/logic unit (ALU), and control unit.*

We know that the system software—or program—can be broken down into a sequence of instructions. Can the system hardware, for both human being and computer, also be broken down into a number of individual blocks? For people we find that it can, and catching a baseball puts five major parts of anatomy (hardware) into play:

- The *eye* (input port) tracks the flight of the ball.
- The *voice* (output port) calls for the ball.
- The *memory* holds the sequence of instructions.
- The *computation and logic area of the brain* (arithmetic/logic unit, or ALU) computes the ball's trajectory.
- The *central nervous system* (control unit) times and sequences the overall process.

As shown in Figure 1.3, these five basic hardware elements are also common to every computer.

The control unit and the ALU are often combined in a single unit known as the *central processing unit*—or simply CPU (the CPU is often a single microprocessor chip, such as the 8080/8085).

7. *The basic processing cycle of a computer system consists of input of data, manipulation of data, and output or display of result.*

Human beings generally interact with their environment in a three-step process: we take in information, manipulate it mentally, and output the result. The overall processing cycle of a computer follows the same pattern: input, process, and output.

BEYOND THE FUNDAMENTAL CONCEPTS

When described in their most basic terms—as we have done in this chapter—the actions that human beings take to catch a baseball or perform other common tasks do not seem complex. It appears we use our marvelous mental machines with little regard for the intricate operations involved. Unfortu-

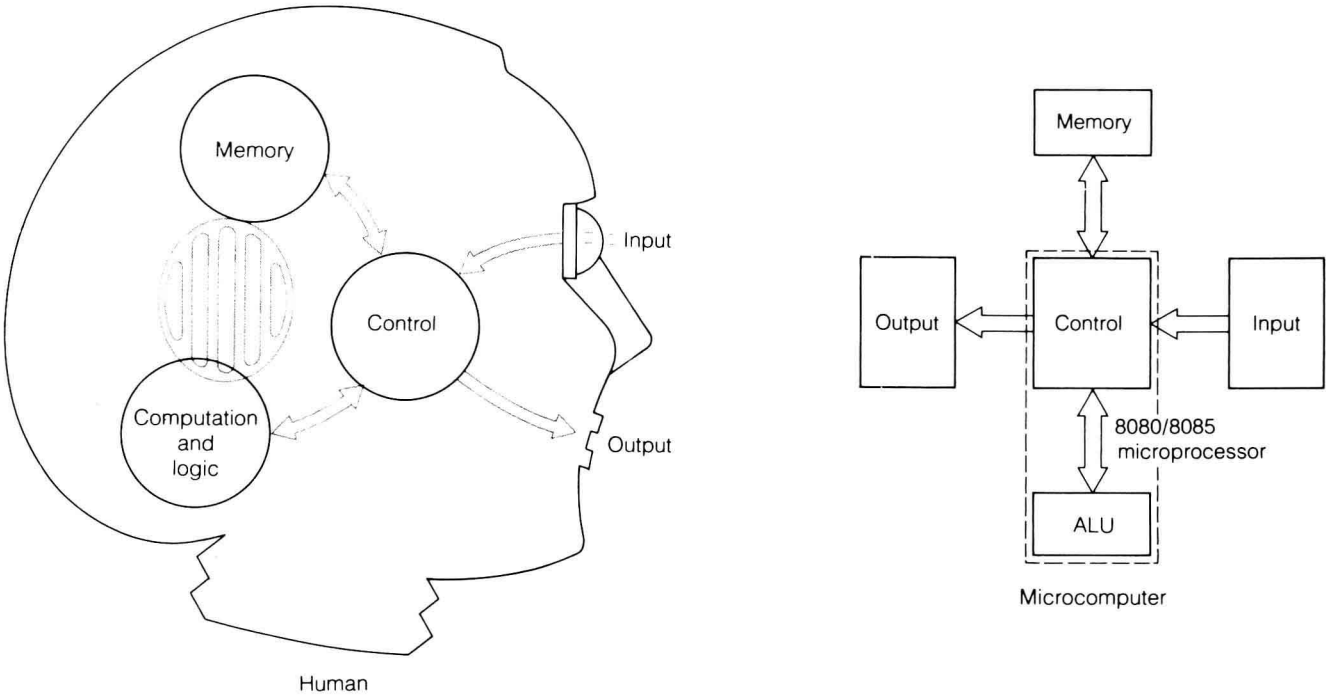


Figure 1.3 The five basic hardware blocks are the same for both human being and computer.