A History of Modern Computing

Paul E. Ceruzzi

The MIT Press
Cambridge, Massachusetts
London, England

This book was set in New Baskerville by Techset Composition Ltd., Salisbury, UK, and was printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Ceruzzi, Paul E.
A history of modern computing / Paul E. Ceruzzi.
p. cm. — (History of computing)
Includes bibliographical references and index.
ISBN 0-262-03255-4 (hardcover : alk. paper)
1. Computers—History. 2. Electronic data processing—History.
I. Title. II. Series.
QA76.17.C47 1998 98-22856
004'.09'045—dc21 CIP

Contents

7 The Personal Computer, 1972–1977

Ready or not, computers are coming to the people.
That's good news, maybe the best since psychedelics.

Those words introduced a story in the fifth anniversary issue of *Rolling Stone*[1] (December 7, 1972). "Spacewar: Fanatic Life and Symbolic Death Among the Computer Bums" was written by Stewart Brand, a lanky Californian who had already made a name for himself as the publisher of the *Whole Earth Catalog*. Brand's resumé was unique, even for an acknowledged hero of the counterculture. At Stanford in the 1960s, he had participated in Defense Department–sponsored experiments with hallucinogenic drugs. In 1968 he had helped Doug Engelbart demonstrate his work on interactive computing at a now-legendary session of the Fall Joint Computer Conference in San Francisco.[2] Brand was no stranger to computers or to the novel ways one might employ them as interactive tools.

Brand was right. Computers did come to the people. The spread of computing to a mass market probably had a greater effect on society than the spread of mind-altering drugs. Personal computing, however, did not arrive in the way that Brand—or almost anyone else—thought it would. The development of personal computing followed a trajectory that is difficult to explain as rational. When trying to describe those years, from 1972 through 1977, one is reminded of Mark Twain's words: "Very few things happen at the right time, and the rest do not happen at all. The conscientious historian will correct these defects."[3] This chapter will examine how computers came "to the people," not as Twain's historian would have written it, but as it really occurred.

What triggered Brand's insight was watching people at the Stanford Artificial Intelligence Laboratory playing a computer game, Spacewar. Spacewar revealed computing as far from the do-not-fold-spindle-ormutilate punched-card environment as one could possibly find. The hardware they were using was not "personal," but the way it was being used was personal: for fun, interactively, with no concern for how many ticks of the processor one was using. That was what people wanted when, two years later, personal computers burst into the market.

Spacewar was running on a PDP-10. In terms of its hardware, a PDP-10 had nothing in common with the personal computers of the next decades.[4] It was large—even DEC's own literature called it a mainframe.[5] It had a 36-bit word length. A full system cost around a half million dollars and easily took up a room of its own. It used discrete transistors and magnetic cores, not integrated circuits, for logic and memory.[6] Still, one can think of the PDP-10 as an ancestor of the personal computer. It was designed from the start to support interactive use. Although its time-sharing abilities were not as ambitious as those of MIT's Project MAC, it worked well. Of all the early time-sharing systems, the PDP-10 best created an illusion that each user was being given the full attention and resources of the computer. That illusion, in turn, created a mental model of what computing could be—a mental model that would later be realized in genuine personal computers.[7]

Chapter 5 discussed the early development of time-sharing and the selection of a General Electric computer for Project MAC at MIT. While that was going on, the MIT Artificial Intelligence Laboratory obtained a DEC PDP-6, the PDP-10's immediate predecessor, for its research (figure 7.1). According to the folklore, MIT students, especially members of the Tech Model Railroad Club, worked closely with DEC on the PDP-6, especially in developing an operating system for it, which would later have an influence on the PDP-10's system software.[8] As a pun on the Compatible Time Sharing System that was running on an IBM mainframe nearby, the students called their PDP-6 system ITS—Incompatible Time Sharing System.[9] The PDP-6 did not have the disk storage necessary to make it a viable time-sharing system and only about twenty were sold. The PDP-10 did have a random-access disk system, which allowed its users direct access to their own personal files.[10] Like other DEC computers, the PDP-10 also allowed users to load personal files and programs onto inexpensive reels of DECtape, which fitted easily into a briefcase.

The feeling that a PDP-10 was one's own personal computer came from its operating system—especially from the way it managed the flow of information to and from the disks or tapes. With MIT's help, DEC supplied a system called "TOPS-10," beginning in 1972. In the

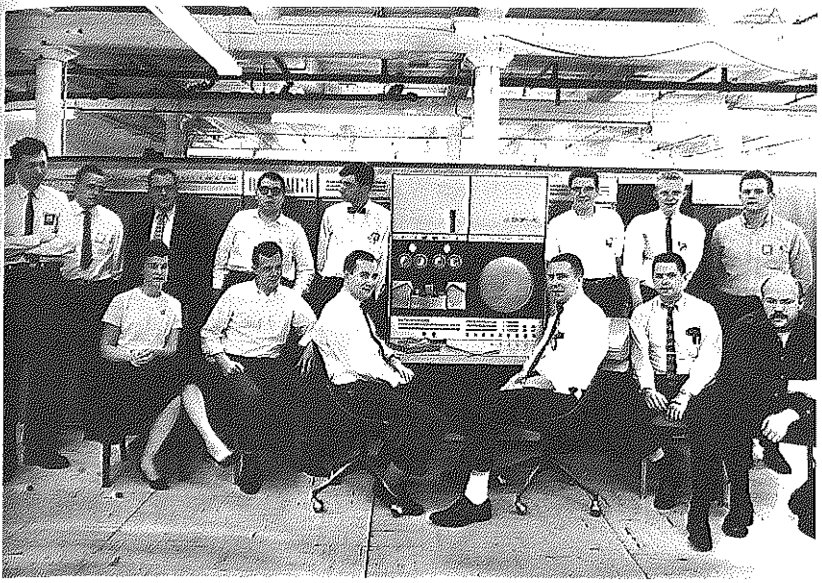

Figure 7.1
One of the most influential computers of all time, the DEC PDP-6, flanked by its creators at the Mill, 1964. C. Gordon Bell is at the left, wearing the sports jacket. The PDP-6 did not sell well but was the prototype for the more successful PDP-10 and DEC System-20. It would have as much of an impact on the course of computing as the much more celebrated PDP-8, also introduced at that time. (*Source:* Digital Equipment Corporation.)

introduction to the TOPS-10 manual, the authors stated, "Our goal has always been that in a properly configured system, each user has the feeling that he owns his portion of the machine for the time he needs to use it."[11] Users could easily create, modify, store, and recall blocks of data from a terminal. The system called these blocks by the already-familiar term, "files." Files were named by one to six characters, followed by a period, then a three-character extension (which typically told what type of file it was, e.g.: xxxxxx.BAS for a program written in BASIC). By typing DIR at a terminal users could obtain a directory of all the files residing on a disk. They could easily send the contents of a file to a desired output device, which typically consisted of a three-letter code, for example, LPT for line printer, or TTY for Teletype.[12]

A small portion of TOPS-10 was always present in core memory. Other programs were stored on the disk and could be called up as necessary. One, called PIP (Peripheral Interchange Program), allowed users to move files in a variety of ways to and from input/output equipment. Another program, TECO (Text Editor and Corrector), allowed users to edit and manipulate text from a terminal. DDT (Dynamic Debugging Tool) allowed users to analyze programs and correct errors without going through the long turnaround times that plagued batch processing.

For PDP-10 users, TOPS-10 was a marvel of simplicity and elegance and gave them the illusion that they were in personal control. TOPS-10 was like a Volkswagen Beetle: basic, simple, and easy to understand and work with.[13] Using a PDP-10 was not only fun but addictive. It was no accident that Brand saw people playing Spacewar on one, or that it was also the computer on which Adventure—perhaps the most long-lasting of all computer games—was written.[14]

On the West Coast another system appeared with similar capabilities, the SDS-940, offered by Scientific Data Systems (SDS) of southern California. The 940 was an extension of a conventional computer, the SDS 930, modified by researchers at Berkeley with support from the Defense Department's Advanced Research Projects Agency. The 940 was more polished than the PDP-10, and it performed well. Still, the PDP-10 seemed to be preferred. At the Xerox Palo Alto Research Center, the legendary lab where so much of personal computing would be created, the staff was encouraged to use SDS machines, since Xerox had just purchased SDS. But the researchers there resisted and instead built a clone of a PDP-10, which they called MAXC—Multiple Access Computer, Xerox—the name a pun on Max Palevsky, the founder of SDS.[15] (Palevsky, after becoming very wealthy from the sale of SDS to Xerox, dabbled in Hollywood movies, politics, and culture—and joined the board of *Rolling Stone*. Palevsky also became a venture capitalist with that money, helping to fund Intel, among other companies.)[16]

For a while, when Wall Street was enamored of anything connected with computers, it was easy to raise money to buy or lease a PDP-10 or SDS-940, and then sell computer time to engineering companies or other customers. Most of these firms were undercapitalized and did not understaud the complexities of what they were selling. Like their counterparts in the electric utility industry, they had to have enough capacity to handle peak loads, in order not to discourage customers. But that meant that during off-peak times they would be wasting unused and expensive computing equipment. The capital requirements necessary to manage the cycles of the business were as large as they were in the electric power business, which had gone through decades of chaos and turmoil before settling down. Ouly a few survived,[17] and even fewer, like

Tymshare of Cupertino, California, did well (although it was sold to McDonnell-Douglas in the early 1980s).[18] Among those many companies, one is worth mentioning, Computer Center Corporation, or C-Cubed, which installed one of the first PDP-10s in the Seattle area in 1968. While it was getting started, it offered a local teenager, Bill Gates, free time on the computer in exchange for helping find and rid the system of bugs. C-Cubed folded in 1970, having given Gates a taste of the potential of interactive computing.[19]

Many of those who had access to these systems saw the future of computing. But the financial troubles of time-sharing companies also showed that it would be difficult to make personal, interactive use widely available. There were attempts to make terminals accessible to the public for free or at low cost—the most famous being the Resource One project in the San Francisco Bay area (partially funded by the *Whole Earth Catalog*). But it did not last, either.[20]

Calculators and Corporate Personal Computer Projects

Economics prevented the spread of computing to the public from the top down—from large mainframes through time-shared terminals. But while those attempts were underway, the underlying technology was advancing rapidly. Could personal computing arrive from the bottom up—from advances in semiconductor electronics?

Many engineers believe that a mental model of the personal computer was irrelevant. They believe that no one invented the personal computer, it simply flowed from advances in semiconductors. Chuck House, an engineer involved with the early Hewlett-Packard calculators, said, "One could uncharitably say that we invented essentially nothing; we simply took all the ideas that were out there and figured out how to implement them cost-effectively." Gordon Bell stated, "The semiconductor density has really been the driving force, and as you reach different density levels, different machines pop out of that in time."[21] To them, inventions are like a piece of fruit that falls to the ground when it is ripe, and the inventor is given credit for doing little more than picking it up. If that were true, one would find a steady progression of machines offering personal, interactive use, as advances in semiconductors made them viable. And these would have come from established firms who had the engineering and manufacturing resources to translate those advances into products.

Products that took advantage of advances in semiconductors did appear on the market. It is worth looking at them to see whether they validate or refute the bottom-up explanation of the PC's invention.

The first electronic computers were of course, operated, as if they were personal computers. Once a person was granted access to a machine (after literally waiting in a queue), he or she had the whole computer to use, for whatever purpose. That gave way to more restricted access, but those at MIT and Lincoln Labs who used the Whirlwind, TX-0, and TX-2 that way never forgot its advantages. In 1962 some of them developed a computer called the LINC, made of Digital Equipment Corporation logic modules and intended for use by a researcher as a personal tool. A demonstration project, funded by the NIH, made sixteen LINCs available to biomedical researchers. DEC produced commercial versions, and by the late 1960s, about 1,200 were in use as personal computers. A key feature of the LINC was its compact tape drive and tapes that one could easily carry around: the forerunner of DECtape. The ease of getting at data on the tape was radically different from the clumsy access of tape in mainframes, and this ease would be repeated with the introduction of floppy-disk systems on personal computers.[22] DEC also marketed a computer that was a combination of a LINC and a PDP-8, for $43,000. Although DECtape soon was offered on nearly all DEC's products, the LINC did not achieve the same kind of commercial success as the PDP-8 and PDP-11 lines of minicomputers.[23]

Advances in chip density first made an impact on personal devices in calculators.[24] For decades there had been a small market for machines that could perform the four functions of arithmetic, plus square root. In the 1950s and 1960s the calculator industry was dominated by firms such as Friden and Marchant in the United States, and Odhner in Europe. Their products were complex, heavy, and expensive.[25] In 1964 Wang Laboratories, a company founded by An Wang, a Chinese immigrant who had worked with Howard Aiken at Harvard, came out with an electronic calculator. The Wang LOCI offered more functions, at a lower cost, than the best mechanical machines. Its successor, the Wang 300, was even easier to use and cheaper, partly because Wang deliberately set the price of the 300 to undercut the competitive mechanical calculators from Friden and others.[26] (Only one or two of the mechanical calculator firms survived the transition to electronics.) A few years later Hewlett-Packard, known for its oscilloscopes and electronic test equipment, came out with the HP-9100A, a calculator selling for just under $5,000. And the Italian firm Olivetti came out with the Programma 101, a $3,500 calculator intended primarily for accounting and statistical work. Besides direct calculation, these machines could also execute a short sequence of steps recorded on magnetic cards.[27] Like the LINC, these calculators used discrete circuits. To display digits, the Wang used "Nixie" tubes, an ingenious tube invented by Burroughs in 1957. HP used a small cathode-ray tube, as might be expected from a company that made oscilloscopes.

By 1970 the first of a line of dramatically cheaper and smaller calculators appeared that used integrated circuits.[28] They were about the size of a paperback book and cost as little as $400. A number of wealthy consumers bought them immediately, but it wasn't until Bowmar advertised a Bowmar Brain for less than $250 for the 1971 Christmas season that the calculator burst into public consciousness.[29] Prices plummeted: under $150 in 1972; under $100 by 1973; under $50 by 1976; finally they became cheap enough to be given away as promotional trinkets.[30] Meanwhile Hewlett-Packard stunned the market in early 1972 with the HP-35, a $400 pocket calculator that performed all the logarithmic and trigonometric functions required by engineers and scientists. Within a few years the slide rule joined the mechanical calculator on the shelves of museums.[31]

Like processed foods, whose cost is mostly in the packaging and marketing, so with calculators: technology no longer determined commercial success. Two Japanese firms with consumer marketing skills, Casio and Sharp, soon dominated. Thirty years after the completion of the half-million dollar ENIAC, digital devices became throw-away commodities. The pioneering calculator companies either stopped making calculators, as did Wang, or went bankrupt, as did Bowmar. Hewlett-Packard survived by concentrating on more advanced and expensive models; Texas Instruments survived by cutting costs.

The commodity prices make it easy to forget that these calculators were ingenious pieces of engineering. Some of them could store sequences of keystrokes in their memory and thus execute short programs. The first of the programmable pocket calculators was Hewlett-Packard's HP-65, introduced in early 1974 for $795 (figure 7.2). Texas Instruments and others soon followed. As powerful as they were, the trade press was hesitant to call them computers, even if Hewlett-Packard introduced the HP-65 as a "personal computer" (possibly the first use of that term in print).[32] Their limited programming was offset by their built-in ability to compute logarithms and trigonometric functions, and to use floating-point arithmetic to ten

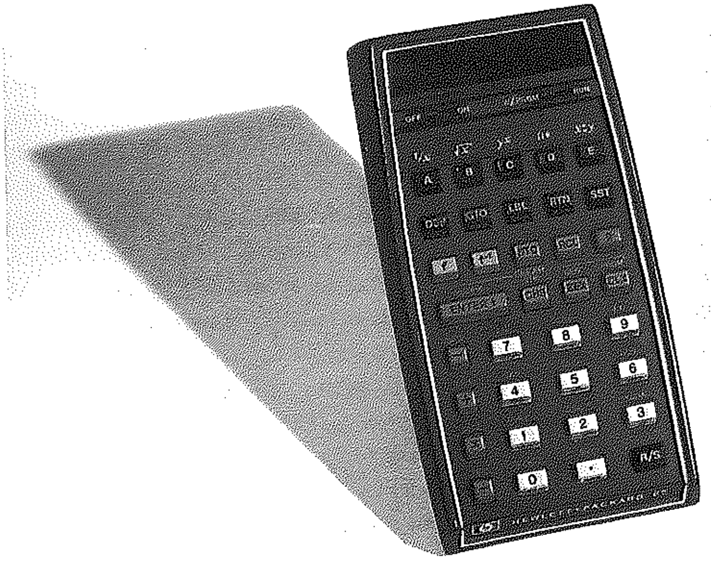

Figure 7.2
HP-65. (*Source:* Smithsonian Institution.)

decimal digits of precision. Few mainframes could do that without custom-written software.

The introduction of pocket programmable calculators had several profound effects on the direction of computing technology. The first was that the calculator, like the Minuteman and Apollo programs of the 1960s, created a market where suppliers could count on a long production run, and thereby gain economies of scale and a low price. As chip density, and therefore capability, increased, chip manufacturers faced the same problem that Henry Ford had faced with his Model T: only long production runs of the same product led to low prices, but markets did not stay static. That was especially true of integrated circuits, which by nature became ever more specialized in their function as the levels of integration increased. (The only exception was in memory chips, which is one reason why Intel was founded to focus on memories). The calculator offered the first consumer market for logic chips that allowed companies to amortize the high costs of designing complex integrated circuits. The dramatic drop in prices of calculators between 1971 and 1976 showed just how potent this force was.[33]

The second effect was just as important. Pocket calculators, especially those that were programmable, unleashed the force of personal creativity and energy of masses of individuals. This force had already created the hacker culture at MIT and Stanford (observed with trepidation by at least one MIT professor).[34] Their story is one of the more colorful among the dry technical narratives of hardware and software design. They and their accomplishments, suitably embellished, have become favorite topics of the popular press. Of course their strange personal habits made a good story, but were they true? Developing system software was hard work, not likely to be done well by a salaried employee, working normal hours and with a family to go home to in the evening. Time-sharing freed all users from the tyranny of submitting decks of cards and waiting for a printout, but it forced some users to work late at night, when the time-shared systems were lightly loaded and thus more responsive.

The assertion that hackers created modern interactive computing is about half-right. In sheer numbers there may never have been more than a few hundred people fortunate enough to be allowed to "hack" (that is, not do a programming job specified by one's employer) on a computer like the PDP-10. By 1975, there were over 25,000 HP-65 programmable calculators in use, each one owned by an individual who could do whatever he or she wished to with it.[35] Who were these people? HP-65 users were not "strange". Nearly all were adult professional men, including civil and electrical engineers, lawyers, financial people, pilots, and so on. Only a few were students (or professors), because they cost $795. Most purchased the HP-65 because they had a practical need for calculation in their jobs. But this was a *personal* machine—one could take it home at night. These users—perhaps 5 or 10 percent of those who owned machines—did not fit the popular notion of hackers as kids with "[t]heir rumpled clothes, their unwashed and unshaven faces, and their uncombed hair."[36] But their passion for programming made them the intellectual cousins of the students in the Tech Model Railroad Club. And their numbers—only to increase as the prices of calculators dropped—were the first indication that personal computing was truly a mass phenomenon.

Hewlett-Packard and Texas Instruments were unprepared for these events. They sold the machines as commodities; they could ill-afford a sales force that could walk a customer through the complex learning process needed to get the most out of one. That was what IBM salesmen were known for—but they sold multimillion dollar mainframes.

Calculators were designed to be easy enough to use to make that unnecessary, at least for basic tasks. What was unexpected was how much more some of those customers wanted to do. Finding little help from the supplier, they turned to one another. Users groups, clubs, newsletters, and publications proliferated.

This supporting infrastructure was critical to the success of personal computing; in the following decade it would become an industry all its own. Many histories of the personal computer emphasize this point; they often cite the role of the Homebrew Computer Club, which met near the Stanford campus in the mid-1970s, as especially important.[37] The calculator users groups were also important, though for different reasons. As the primitive first personal computers like the Altair gave way to more complete systems, a number of calculator owners purchased one of them as well. In the club newsletters there were continuous discussions of the advantages and drawbacks of each—the one machine having the ability to evaluate complex mathematical expressions with ease, the other more primitive but *potentially* capable of doing all that and more.[38] There was no such thing as a typical member of the Homebrew Computer Club, although calculator owners tended to be professionals whose jobs required calculation during the day, and who thought of other uses at night. Many of them were bitten by the PC bug; at the same time they took a show-me attitude toward the computer. Could you rely on one? Could you use one to design a radar antenna? Could it handle a medium-sized mailing list? Was the personal computer a serious machine? At first the answers were, "not yet," but gradually, with some firm prodding by this community, the balance shifted. Groups like the Homebrew Computer Club emphasized the "personal" in personal computer; calculator users emphasized the word computer.

Ever since time-sharing and minicomputers revealed an alternative to mainframe computing, there have been prophets and evangelists who raged against the world of punched cards and computer rooms, promising a digital paradise of truly interactive tools. The most famous was Ted Nelson, whose self-published book *Computer Lib* proclaimed (with a raised fist on the cover): "You can and must understand computers *now*."[39] By 1974 enough of these dreams had become real that the specific abilities—and limits—of actual "dream machines" (the alternate title to Nelson's book) had to be faced. Some of the dreamers, including Nelson, were unable to make the transition. They dismissed the pocket calculator. They thought it was puny, too cheap, couldn't do graphics, wasn't a "von Neumann machine," and so on.[40] For them, the dream machine was better, even if (or because) it was unbuilt.[41] By 1985 there would be millions of IBM Personal Computers and their copies in the offices and homes of ordinary people. These computers would use a processor that was developed for other purposes, and adapted for the personal computer almost by accident. But they would be real and a constant source of inspiration and creativity to many who used them, as well as an equal source of frustration for those who knew how much better they could be.

The Microprocessor

Calculators showed what integrated circuits could do, but they did not open up a direct avenue to personal interactive computing. The chips used in them were too specialized for numerical calculation to form a basis for a general-purpose computer. Their architecture was ad-hoc and closely guarded by each manufacturer. What was needed was a set of integrated circuits—or even a single integrated circuit—that incorporated the basic architecture of a general-purpose, stored-program computer.[42] Such a chip, called a "microprocessor," did appear.

In 1964 Gordon Moore, then of Fairchild and soon a cofounder of Intel, noted that from the time of the invention of integrated circuits in 1958, the number of circuits that one could place on a single integrated circuit was doubling every year.[43] By simply plotting this rate on a piece of semi-log graph paper, "Moore's Law" predicted that by the mid 1970s one could buy a chip containing logic circuits equivalent to those used in a 1950s-era mainframe. (Recall that the UNIVAC I had about 3,000 tubes, about the same number of active elements contained in the first microprocessor discussed below.) By the late 1960s transistor-transistor logic (TTL) was well established, but a new type of semiconductor called metal-oxide semiconductor (MOS), emerged as a way to place even more logic elements on a chip.[44] MOS was used by Intel to produce its pioneering 1103 memory chip, and it was a key to the success of pocket calculators. The chip density permitted by MOS brought the concept of a computer-on-a-chip into focus among engineers at Intel, Texas Instruments, and other semiconductor firms. That did not mean that such a device was perceived as useful. If it was generally known that enough transistors could be placed on a chip to make a computer, it was also generally believed that the market for such a chip was so low that its sales would never recoup the large development costs required.[45]

By 1971 the idea was realized in silicon. Several engineers deserve credit for the invention. Ted Hoff, an engineer at Intel, was responsible for the initial concept, Federico Faggin of Intel deserves credit for its realization in silicon, and Gary Boone of Texas Instruments designed similar circuits around that time. In 1990, years after the microprocessor became a household commodity and after years of litigation, Gil Hyatt, an independent inventor from La Palma, California, received a patent on it. Outside the courts he has few supporters, and recent court rulings may have invalidated his claim entirely.[46]

The story of the microprocessor's invention at Intel has been told many times.[47] In essence, it is a story encountered before: Intel was asked to design a special-purpose system for a customer. It found that by designing a general-purpose computer and using software to tailor it to the customer's needs, the product would have a larger market.

Intel's customer for this circuit was Busicom, a Japanese company that was a top seller of hand-held calculators. Busicom sought to produce a line of products with different capabilities, each aimed at a different market segment. It envisioned a set of custom-designed chips that incorporated the logic for the advanced mathematical functions. Intel's management assigned Marcian E. Hoff, who had joined the company in 1968 (Intel's twelfth employee), to work with Busicom.

Intel's focus had always been on semiconductor memory chips. It had shied away from logic chips like those suggested by Busicom, since it felt that markets for them were limited. Hoff's insight was to recognize that by designing fewer logic chips with more general capabilities, one could satisfy Busicom's needs elegantly. Hoff was inspired by the PDP-8, which had a very small set of instructions, but which its thousands of users had programmed to do a variety of things. He also recalled using an IBM 1620, a small scientific computer with an extremely limited instruction set that nevertheless could be programmed to do a lot of useful work.

Hoff proposed a logic chip that incorporated more of the concepts of a general-purpose computer (figure 7.3). A critical feature was the ability to call up a subroutine, execute it, and return to the main program as needed.[48] He proposed to do that with a register that kept track of where a program was in its execution and saved that status when interrupted to perform a subroutine. Subroutines themselves could be interrupted, with return addresses stored on a "stack": an arrangement of registers that automatically retrieved data on a last-in-first-out basis.[49]

With this ability, the chip could carry out complex operations stored as subroutines in memory, and avoid having those functions perma-

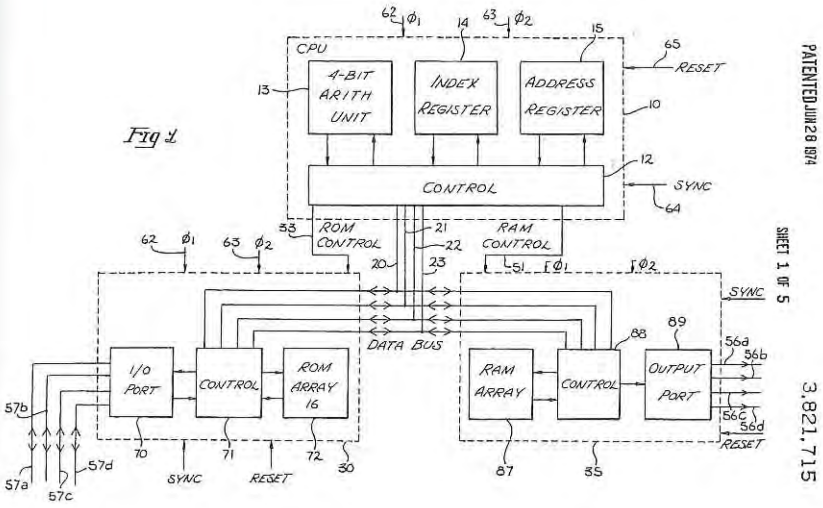

Figure 7.3
(*top*) Patent for a "Memory System for a Multi-Chip Digital Computer," by M. E. Hoff, Stanley Mazor, and Federico Faggin of Intel. The patent was not specifically for a "computer on a chip," but note that all the functional blocks found in the processor of a stored-program computer are shown in this drawing. (*bottom*) Intel 8080. (*Source:* Smithsonian Institution.)

nently wired onto the chip. Doing it Hoff's way would be slower, but in a calculator that did not matter, since a person could not press keys that fast anyway. The complexity of the logic would now reside in software stored in the memory chips, so one was not getting something for nothing. But Intel was a memory company, and it knew that it could provide memory chips with enough capacity. As an added inducement, sales of the logic chips would mean more sales of its bread-and-butter memories.

That flexibility meant that the set of chips could be used for many other applications besides calculators. Busicom was in a highly competitive and volatile market, and Intel recognized that. (Busicom eventually went bankrupt.) Robert Noyce negotiated a deal with Busicom to provide it with chips at a lower cost, giving Intel in return the right to market the chips to other customers for noncalculator applications. From these unsophisticated negotiations with Busicom, in Noyce's words, came a pivotal moment in the history of computing.[50]

The result was a set of four chips, first advertised in a trade journal in late 1971, which included "a microprogrammable computer on a chip!"[51] That was the 4004, on which one found all the basic registers and control functions of a tiny, general-purpose stored-program computer. The other chips contained a read-only memory (ROM), random-access memory (RAM), and a chip to handle output functions. The 4004 became the historical milestone, but the other chips were important as well, especially the ROM chip that supplied the code that turned a general-purpose processor into something that could meet a customer's needs. (Also at Intel, a team led by Dov Frohman developed a ROM chip that could be easily reprogrammed and erased by exposure to ultraviolet light. Called an EPROM (erasable programmable read-only memory) and introduced in 1971, it made the concept of system design using a microprocessor practical.)[52]

The detailed design of the 4004 was done by Stan Mazor. Federico Faggin was also crucial in making the concept practical. Masatoshi Shima, a representative from Busicom, also contributed. Many histories of the invention give Hoff sole credit; all players, including Hoff, now agree that that is not accurate. Faggin left Intel in 1974 to found a rival company, Zilog. Intel, in competition with Zilog, felt no need to advertise Faggin's talents in its promotional literature, although Intel never showed any outward hostility to its ex-employee.[53] The issue of whom to credit reveals the way many people think of invention: Hoff had the idea of putting a general-purpose computer on a chip, Faggin and the others "merely" implemented that idea in silicon. At the time, Intel was not sure what it had invented either: Intel's patent attorney resisted Hoff's desire at the time to patent the work as a "computer."[54] Intel obtained two patents on the 4004, covering its architecture and implementation; Hoff's name appears on only one of them. (That opened the door to rival claims for patent royalties from TI, and eventually Gil Hyatt.)

The 4004 worked with groups of four bits at a time—enough to code decimal digits but no more. At almost the same time as the work with

Busicom, Intel entered into a similar agreement with Computer Terminal Corporation (later called Datapoint) of San Antonio, Texas, to produce a set of chips for a terminal to be attached to mainframe computers. Again, Mazor and Hoff proposed a microprocessor to handle the terminal's logic. Their proposed chip would handle data in 8-bit chunks, enough to process a full byte at a time. By the time Intel had completed its design, Datapoint had decided to go with conventional TTL chips. Intel offered the chip, which they called the 8008, as a commercial product in April 1972.[55]

In late 1972, a 4-bit microprocessor was offered by Rockwell, an automotive company that had merged with North American Aviation, maker of the Minuteman Guidance System. In 1973 a half dozen other companies began offering microprocessors as well. Intel responded to the competition in April 1974 by announcing the 8080, an 8-bit chip that could address much more memory and required fewer support chips than the 8008. The company set the price at $360—a somewhat arbitrary figure, as Intel had no experience selling chips like these one at a time. (Folklore has it that the $360 price was set to suggest a comparison with the IBM System/360.)[56] A significant advance over the 8008, the 8080 could execute programs written for the other chip, a compatibility that would prove crucial to Intel's dominance of the market. The 8080 was the first of the microprocessors whose instruction set and memory addressing capability approached those of the minicomputers of the day.[57]

From Microprocessor to Personal Computer

There were now, in early 1974, two converging forces at work. From one direction were the semiconductor engineers with their ever-more-powerful microprocessors and ever-more-capacious memory chips. From the other direction were users of time-sharing systems, who saw a PDP-10 or XDS 940 as a basis for public access to computing. When these forces met in the middle, they would bring about a revolution in personal computing.

They almost did not meet. For the two years between Brand's observation and the appearance of the Altair, the two forces were rushing past one another. The time-sharing systems had trouble making money even from industrial clients, and the public systems like Community Memory were also struggling. At the other end, semiconductor companies did not think of their products as a possible basis for a personal computer.

A general-purpose computer based on a microprocessor did appear in 1973. In May of that year Thi T. Truong, an immigrant to France from Viet Nam, had his electronics company design and build a computer based on the Intel 8008 microprocessor. The MICRAL was a rugged and well-designed computer, with a bus architecture and internal slots on its circuit board for expansion. A base model cost under $2,000, and it found a market replacing minicomputers for simple control operations. Around two thousand were sold in the next two years, none of them beyond an industrial market.[58] It is regarded as the first microprocessor-based computer to be sold in the commercial marketplace. Because of the limitations of the 8008, its location in France, and above all, the failure by its creators to see what it "really" was, it never broke out of its niche as a replacement for minicomputers in limited industrial locations.

The perception of the MICRAL as something to replace the mini was echoed at Intel as well. Intel's mental model of its product was this: an *industrial* customer bought an 8080 and wrote specialized software for it, which was then burned into a read-only-memory to give a system with the desired functions. The resulting inexpensive product (no longer programmable) was then put on the market as an embedded controller in an industrial system. A major reason for that mental model was the understanding of how hard it was to program a microprocessor. It seemed absurd to ask untrained consumers to program when Intel's traditional customers, hardware designers, were themselves uncomfortable with programming.

With these embedded uses in mind, microprocessor suppliers developed educational packages intended to ease customers into system design. These kits included the microprocessor, some RAM and ROM chips, and some other chips that handled timing aud control, all mounted on a printed circuit board. They also included written material that gave a tutorial on how to program the system. This effort took Intel far from its core business of making chips, but the company hoped to recoup the current losses later on with volume sales of components.[59] These kits were sold for around $200 or given away to engineers who might later generate volume sales.

Intel and the others also built more sophisticated "Development Systems," on which a customer could actually test the software for an application (figure 7.4). These were fully assembled products that sold

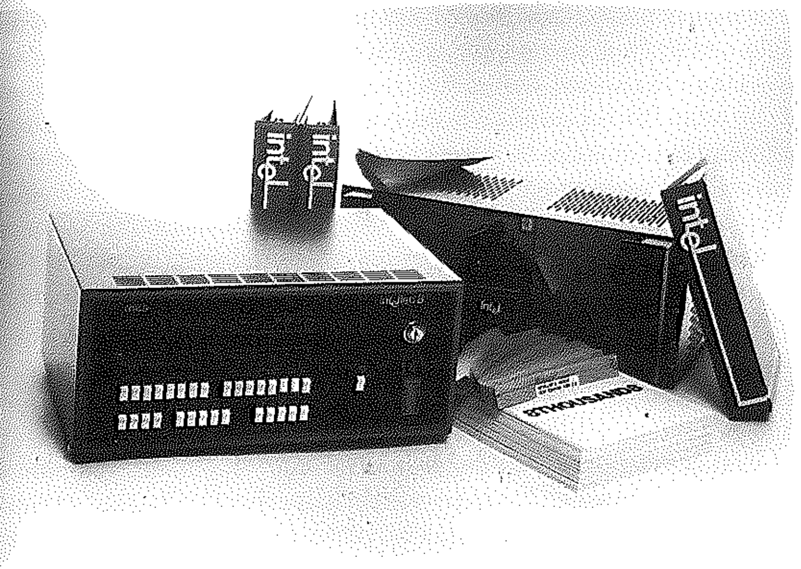

Figure 7.4
Intellec-8 Development System. This was, in fact, a general-purpose computer, but Intel did not market it as such. Intel intended that customers buy them to assist in writing and debugging microprocessor software that would go into embedded systems. A few were purchased and used as alternatives to minicomputers. (*Source:* Intel.)

for around $10,000. To use these systems, customers also needed specialized software that would allow them to write programs using a language like FORTRAN, and then "cross-compile" it for the microprocessor—that is, from the FORTRAN program generate machine code, not for the computer on which it was written, but for the microprocessor. The company hired Gary Kildall, an instructor at the Naval Postgraduate School in Monterey, California, to develop a language based on IBM's PL/1.[60] He called it PL/M, and in 1973 Intel offered it to customers. Initially this software was intended to be run on a large mainframe, but it was soon available for minicomputers, and finally to microprocessor-based systems. In 1974 Intel offered a development system, the Intellec 4, which included its own resident PL/M compiler (i.e., one did not need a mainframe or a mini to compile the code).[61] A similar Intellec-8 introduced the 8-bit microprocessors.

With these development systems, Intel had in fact invented a personal computer. But the company did not realize it. These kits were not marketed as the functional computers they were. Occasionally someone bought one of these systems and used it in place of a minicomputer, but Intel neither supported that effort nor recognized its potential.[62] Intel and the other microprocessor firms made money selling these development systems—for some they were very profitable—but the goal was to use them as a lever to open up volume purchases of chips. The public could not buy one. The chip suppliers were focused on the difficulties in getting embedded systems to do useful work; they did not think that the public would be willing to put up with the difficulties of programming just to own their own computer.

Role of Hobbyists

Here is where the electronics hobbyists and enthusiasts come in. Were it not for them, the two forces in personal computing might have crossed without converging. Hobbyists, at that moment, were willing to do the work needed to make microprocessor-based systems practical.

This community had a long history of technical innovation—it was radio amateurs, for example, who opened up the high-frequency radio spectrum for long-distance radio communications after World War I. After World War II, the hobby expanded beyond amateur radio to include high-fidelity music reproduction, automatic controls, and simple robotics. A cornucopia of war surplus equipment from the U.S. Army Signal Corps found its way into individual hands, further fueling the phenomenon. (A block in lower Manhattan known as "Radio Row," where the World Trade Center now stands, was a famous source of surplus electronic gear.)[63] The shift from vacuum tubes to integrated circuits made it harder for an individual to build a circuit on a breadboard at home, but inexpensive TTL chips now contained whole circuits themselves.[64] As the hobby evolved rapidly from analog to digital applications, this group supplied a key component in creating the personal computer: It provided an infrastructure of support that neither the computer companies nor the chip makers could.

This infrastructure included a variety of electronics magazines. Some were aimed at particular segments, for example, *QST* for radio amateurs. Two of them, *Popular Electronics* and *Radio-Electronics*, were of general interest and sold at newsstands; they covered high-fidelity audio, shortwave radio, television, and assorted gadgets for the home and car. Each issue typically had at least one construction project. For these projects the magazine would make arrangements with small electronics companies to supply a printed circuit board, already etched and drilled, as well as specialized components that readers might have difficulty finding locally. By scanning the back issues of these magazines we can trace how hobbyists moved from analog to digital designs.

A machine called the Kenbak-1, made of medium and small-scale integrated circuits, was advertised in the September 1971 issue of *Scientific American.* The advertisement called it suitable for "private individuals," but it was really intended for schools. The Kenbak may be the first personal computer, but it did not use a microprocessor, and its capabilities were quite limited.

The Scelbi-8H was announced in a tiny advertisement in the back of the March 1974 issue of *QST.* It used an Intel 8008, and thus may be the first microprocessor-based computer marketed to the public. According to the advertisement, "Kit prices for the new Scelbi-8H mini-computer start as low as $440!"[65] It is not known how many machines Scelbi sold, but the company went on to play an important part in the early personal computer phenomenon.[66]

In July 1974, *Radio-Electronics* announced a kit based on the Intel 8008, under the headline "Build the Mark-8: Your Personal Minicomputer."[67] The project was much more ambitious than what typically appeared in that magazine. The article gave only a simple description and asked readers to order a separate, $5.00 booklet for detailed instructions. The Mark-8 was designed by Jonathan Titus of Virginia Polytechnic University in Blacksburg. The number of machines actually built may range in the hundreds, although the magazine reportedly sold "thousands" of booklets. At least one Mark-8 users club sprang up, in Denver, whose members designed an ingenious method of storing programs on an audio cassette recorder.[68] Readers were directed to a company in Englewood, New Jersey, that supplied a set of circuit boards for $47.50, and to Intel for the 8008 chip (for $120.00). The Mark-8's appearance in *Radio-Electronics* was a strong factor in the decision by its rival *Popular Electronics* to introduce the Altair kit six months later.[69]

These kits were just a few of many projects described in the hobbyist magazines. They reflected a conscious effort by the community to bring digital electronics, with all its promise and complexity, to amateurs who were familiar only with simpler radio or audio equipment. It was not an easy transition: construction of both the Mark-8 and the TV-typewriter (described next) was too complex to be described in a magazine article; readers had to order a separate booklet to get complete plans. *Radio-Electronics* explained to its readers that "[w]e do not intend to do an article this way as a regular practice."[70] Although digital circuits were more complex than what the magazine had been handling, it recognized that the electronics world was moving in that direction and that its readers wanted such projects.

Other articles described simpler digital devices—timers, games, clocks, keyboards, and measuring instruments—that used inexpensive TTL chips. One influential project was the TV-Typewriter, designed by Don Lancaster and published in *Radio-Electronics* in September 1973. This device allowed readers to display alphanumeric characters, encoded in ASCII, on an ordinary television set. It presaged the advent of CRT terminals as the primary input-output device for personal computers—one major distinction between the PC culture and that of the minicomputer, which relied on the Teletype. Lee Felsenstein called the TV-Typewriter "the opening shot of the computer revolution."[71]

Altair

1974 was the *annus mirabilis* of personal computing. In January, Hewlett-Packard introduced its HP-65 programmable calculator. That summer Intel announced the 8080 microprocessor. In July, *Radio-Electronics* described the Mark-8. In late December, subscribers to *Popular Electronics* received their January 1975 issue in the mail, with a prototype of the "Altair" minicomputer on the cover (figure 7.5), and an article describing how readers could obtain one for less than $400. This announcement ranks with IBM's announcement of the System/360 a decade earlier as one of the most significant in the history of computing. But what a difference a decade made: the Altair was a genuine personal computer.

H. Edward Roberts, the Altair's designer, deserves credit as the inventor of the personal computer. The Altair was a capable, inexpensive computer designed around the Intel 8080 microprocessor. Although calling Roberts the inventor makes sense only in the context of all that came before him, including the crucial steps described above, he does deserve the credit. Mark Twain said that historians have to rearrange past events so they make more sense. If so, the invention of the personal computer at a small model-rocket hobby shop in Albuquerque cries out for some creative rearrangement. Its utter improbability and unpredictability have led some to credit many other places with the invention, places that are more sensible, such as the Xerox Palo Alto Research Center, or Digital Equipment Corporation, or even IBM. But Albuquer-

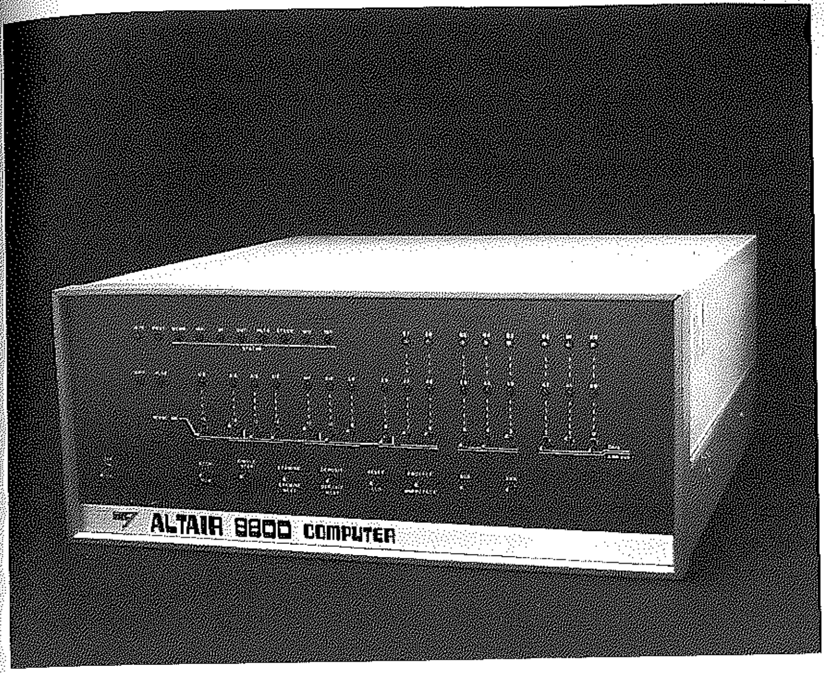

Figure 7.5
MITS Altair 8800 Computer. The front panel was copied from the Data General Nova. The machine shown in this photograph was one of the first produced and was owned by Forrest Mims, an electronics hobbyist and frequent contributor to *Popular Electronics*, who had briefly worked at MITS. (*Source:* Smithsonian Institution.)

que it was, for it was only at MITS that the technical and social components of personal computing converged.

Consider first the technical. None of the other hobbyist projects had the impact of the Altair's announcement. Why? One reason was that it was designed and promoted as a capable minicomputer, as powerful as those offered by DEC or Data General. The magazine article, written by Ed Roberts and William Yates, makes this point over and over: "a full-blown computer than can hold its own against sophisticated minicomputers"; "not a 'demonstrator' or a souped-up calculator"; or "performance competes with current commercial minicomputers."[72] The physical appearance of the Altair computer suggested its minicomputer lineage. It looked like the Data General Nova: it had a rectangular metal case, a front panel of switches that controlled the contents of internal registers, and small lights indicating the presence of a binary one or zero. Inside the Altair's case, there was a machine built mainly of TTL integrated circuits (except for the microprocessor, which was a MOS device), packaged in dual-in-line packages, soldered onto circuit boards. Signals and power traveled from one part of the machine to another on a bus. The Altair used integrated circuits, not magnetic cores, for its primary memory. The *Popular Electronics* cover called the Altair the "world's first minicomputer kit"; except for its use of a microprocessor, that accurately described its physical construction and design.[73]

But the Altair as advertised was ten times cheaper than minicomputers were in 1975. The magazine offered an Altair for under $400 as a kit, and a few hundred more already assembled. The magazine cover said that readers could "save over $1,000." In fact, the cheapest PDP-8 cost several thousand dollars. Of course, a PDP-8 was a fully assembled, operating computer that was considerably more capable than the basic Altair, but that did not really matter in this case. (Just what one got for $400 will be discussed later.) The low cost resulted mainly from its use of the Intel 8080 microprocessor, just introduced. Intel had quoted a price of $360 for small quantities of 8080s, but Intel's quote was not based on a careful analysis of how to sell the 8080 to this market. MITS bought them for only $75 each.[74]

The 8080 had more instructions and was faster and more capable than the 8008 that the Mark-8 and Scelbi-8 used. It also permitted a simpler design since it required only six instead of twenty supporting chips to make a functional system. Other improvements over the 8008 were its ability to address up to 64 thousand bytes of memory (vs. the 8008's 16 thousand), and its use of main memory for the stack, which permitted essentially unlimited levels of subroutines instead of the 8008's seven levels.

The 8080 processor was only one architectural advantage the Altair had over its predecessors. Just as important was its use of an open bus.[75] According to folklore, the bus architecture almost did not happen. After building the prototype Altair, Roberts photographed it and shipped it via Railway Express to the offices of *Popular Electronics* in New York. Railway Express, a vestige of an earlier American industrial revolution, was about to go bankrupt; it lost the package. The magazine cover issue showed the prototype, with its light-colored front panel and the words "Altair 8800" on the upper left. That machine had a set of four large circuit boards stacked on top of one another, with a wide ribbon cable carrying 100 lines from one board to another. After that machine was lost, Robert redesigned the Altair. He switched to a larger deep blue cabinet and discarded the 100-wire ribbon cable. In the new design, wires connected to a rigid backplane carried the signals from one board to another. That allowed hobbyists to add a set of connectors that could accept other cards besides the initial four.[76]

The $400 kit came with only two cards to plug into the bus: those two, plus a circuit board to control the front panel and the power supply, made up the whole computer. The inside looked quite bare. But laboriously soldering a set of wires to an expansion chassis created a full set of slots into which a lot of cards could be plugged. MITS was already designing cards for more memory, I/O and other functions.

Following the tradition established by Digital Equipment Corporation, Roberts did not hold specifications of the bus as a company secret. That allowed others to design and market cards for the Altair. That decision was as important to the Altair's success as its choice of an 8080 processor. It also explains one of the great ironies of the Altair, that it inaugurated the PC era although it was neither reliable nor very well-designed. Had it not been possible for other companies to offer plug-in cards that improved on the original MITS design, the Altair might have made no greater impact than the Mark-8 had. The bus architecture also led to the company's demise a few years later, since it allowed other companies to market compatible cards and, later, compatible computers. But by then the floodgates had opened. If MITS was unable to deliver on its promises of making the Altair a serious machine (though it tried), other companies would step in. MITS continued developing plug-in cards and peripheral equipment, but the flood of orders was too much for the small company.

So while it was true that for $400 hobbyists got very little, they could get the rest—or design and build the rest. Marketing the computer as a bare-bones kit offered a way for thousands of people to bootstrap their way into the computer age, at a pace that they, not a computer company, could control.

Assembling the Altair was much more difficult than assembling other electronics kits, such as those sold by the Heath Company or Dynaco. MITS offered to sell "completely assembled and tested" computers for $498, but with such a backlog of orders, readers were faced with the choice of ordering the kit and getting something in a couple of months, or ordering the assembled computer and perhaps waiting a year or more.[77] Most ordered the kit and looked to one another for support in finding the inevitable wiring errors and poorly soldered connections that they would make. The audience of electronics hobbyists, at whom the magazine article was aimed, compared the Altair not to the simple Heathkits, but to building a computer from scratch, which was almost impossible: not only was it hard to design a computer, it was impossible to obtain the necessary chips. Chips were inexpensive, but only if they were purchased in large quantities, and anyway, most semiconductor firms had no distribution channels set up for single unit or retail sales. Partly because of this, customers felt, rightly, that they were getting an incredible bargain.

The limited capabilities of the basic Altair, plus the loss of the only existing Altair by the time the *Popular Electronics* article appeared, led to the notion that it was a sham, a "humbug," not a serious product at all.[78] The creators of the Altair fully intended to deliver a serious computer whose capabilities were on a par with minicomputers then on the market. Making those deliveries proved to be a lot harder than they anticipated. Fortunately, hobbyists understood that. But there should be no mistake about it: the Altair was real.

MITS and the editors of *Popular Electronics* had found a way to bring the dramatic advances in integrated circuits to individuals. The first customers were hobbyists, and the first thing they did with these machines, once they got them running, was play games.[79] Roberts was trying to sell it as a machine for serious work, however. In the *Popular Electronics* article he proposed a list of twenty-three applications, none of them games.[80] Because it was several years before anyone could supply peripheral equipment, memory, and software, serious applications were rare at first. That, combined with the primitive capabilities of other machines like the Mark-8, led again to an assumption that the Altair was not a serious computer. Many of the proposed applications hinted at in the 1975 article were eventually implemented. Years later one could still find an occasional Altair (or more frequently, an Altair clone) embedded into a system just like its minicomputer cousins.

The next three years, from January 1975 through the end of 1977, saw a burst of energy and creativity in computing that had almost no equal in its history. The Altair had opened the floodgates, even though its shortcomings were clear to everyone. One could do little more than get it to blink a pattern of lights on the front panel. And even that was not easy: one had to flick the toggle switches for each program step, then deposit that number into a memory location, then repeat that for the next step, and so on—hopefully the power did not go off while this was going on—until the whole program (less than 256 bytes long!) was in memory. Bruised fingers from flipping the small toggle switches were the least of the frustrations. In spite of all that, the bus architecture meant that other companies could design boards to remedy each of these shortcomings, or even design a copy of the Altair itself, as IMSAI and others did.[81]

But the people at MITS and their hangers-on created more than just a computer. This $400 computer inspired the extensive support of user groups, informal newsletters, commercial magazines, local clubs, conventions, and even retail stores. This social activity went far beyond traditional computer user groups, like SHARE for IBM or DECUS for Digital. Like the calculator users groups, these were open and informal, and offered more to the neophyte. All of this sprang up with the Altair, and many of the publications and groups lived long after the last Altair computer itself was sold.

Other companies, beginning with Processor Technology, soon began offering plug-in boards that gave the machine more memory. Another board provided a way of connecting the machine to a Teletype, which allowed fingers to heal. But Teletypes were not easy to come by—an individual not affiliated with a corporation or university could only buy one secondhand, and even then they were expensive. Before long, hobbyists-led small companies began offering ways of hooking up a television set and a keyboard (although Don Lancaster's TV Typewriter was not the design these followed). Tbe board that connected to the Teletype sent data serially—one bit at a time; another board was designed that sent out data in parallel, for connection to a liue printer that minicomputers used, although like the Teletype these were expensive and hard to come by.[82]

The Altair lost its data when the power was shut off, but before long MITS designed an interface that put out data as audio tones, to store programs on cheap audio cassettes. A group of hobbyists met in Kansas City in late 1975 and established a "Kansas City Standard" for the audio tones stored on cassettes, so that programs could be exchanged from one computer to another.[83] Some companies brought out inexpensive paper tape readers that did not require the purchase of a Teletype. Others developed a tape cartridge like the old 8-track audio systems, which looped a piece of tape around and around. Cassette storage was slow and cumbersome—users usually had to record several copies of a program and make several tries before successfully loading it into the computer. Inadequate mass storage limited the spread of PCs until the "floppy" disk was adapted.

The floppy was invented by David L. Noble at IBM for a completely different purpose. When IBM introduced the System/370, which used semiconductor memory, it needed a way to store the computer's initial control program, as well as to hold the machine's microprogram. That had not been a problem for the System/360, which used magnetic cores that held their contents when the power was switched off. From this need came the 8-inch diameter flexible diskette, which IBM announced in 1971.[84] Before long, people recognized that it could be used for other purposes besides the somewhat limited one for which it had been invented. In particular, Alan Shugart, who had once worked for IBM, recognized that the floppy's simplicity and low cost made it the ideal storage medium for low-cost computer systems.[85] Nevertheless, floppy drives were rare in the first few years of personal computing. IBM's hardware innovation was not enough; there had to be an equivalent innovation in system software to make the floppy practical. Before that story is told, we shall look first at the more immediate issue of developing a high-level language for the PC.

Software: BASIC

The lack of a practical mass storage device was one of two barriers that blocked the spread of personal, interactive computing. The other was a way to write applications software. By 1977 two remarkable and influential pieces of software—Microsoft BASIC and the CP/M Operating System—overcame those barriers.

In creating the Altair, Ed Roberts had to make a number of choices: what processor to use, the design of the bus (even whether to use a bus at all), the packaging, and so on. One such decision was the choice of a programming language. Given the wide acceptance of BASIC it is hard to imagine that there ever was a choice, but there was. BASIC was not invented for small computers. The creators of BASIC abhorred the changes others made to shoehorn the language onto systems smaller than a mainframe. Even in its mainframe version, BASIC had severe limitations—on the numbers and types of variables it allowed, for example. In the view of academic computer scientists, the versions of BASIC developed for minicomputers were even worse—full of ad hoc patches and modifications. Many professors disparaged BASIC as a toy language that fostered poor programming habits, and they refused to teach it. Serious programming was done in FORTRAN—an old and venerable but still capable language.

If, in 1974, one asked for a modern, concise, well-designed language to replace FORTRAN, the answer might have been APL, an interactive language invented at IBM by Kenneth Iverson in the early 1960s. A team within IBM designed a personal computer in 1973 that supported APL, the "SCAMP," although a commercial version of that computer sold poorly.[86] Or PL/I: IBM had thrown its resources into this language, which it hoped would replace both FORTRAN and COBOL. Gary Kildall chose a subset of PL/I for the Intel microprocessor development kit.

BASIC's strength was that it was easy to learn. More significant, it already had a track record of running on computers with limited memory. Roberts stated that he had considered FORTRAN and APL, before he decided the Altair was to have BASIC.[87]

William Gates III was born in 1955, at a time when work on FORTRAN was just underway. He was a student at Harvard when the famous cover of *Popular Electronics* appeared describing the Altair. According to one biographer, his friend Paul Allen saw the magazine and showed it to Gates, and the two immediately decided that they would write a BASIC compiler for the machine.[88] Whether it was Gates's or Roberts's decision to go with BASIC for the Altair, BASIC it was (figure 7.6).

In a newsletter sent out to Altair customers, Gates and Allen stated that a version of BASIC that required only 4K bytes of memory would be available in June 1975, and that more powerful versions would be available soon after. The cost, for those who also purchased Altair memory boards, was $60 for 4K BASIC, $745 for 8K, and $150 for "extended" BASIC (requiring disk or other mass storage). Those who wanted the language to run on another 8080-based system had to pay $500.[89]

In a burst of energy, Gates and Allen, with the help of Monte Davidoff, wrote not only a BASIC that fit into very little memory; they wrote a BASIC with a lot of features and impressive performance. The language was true to its Dartmouth roots in that it was easy to learn. It broke with those roots by providing a way to move from BASIC commands to instructions written in machine language. That was primarily through a USR command, which was borrowed from software written for DEC minicomputers (where the acronym stood for user service routine).[90] A programmer could even directly put bytes into or pull data out of specific memory locations, through the PEEK and POKE commands—which would have caused havoc on the time-shared Dartmouth system. Like USR, these commands were also derived from prior work done by DEC programmers, who came up with them for a time-sharing system they wrote in BASIC for the PDP-11. Those commands allowed users to

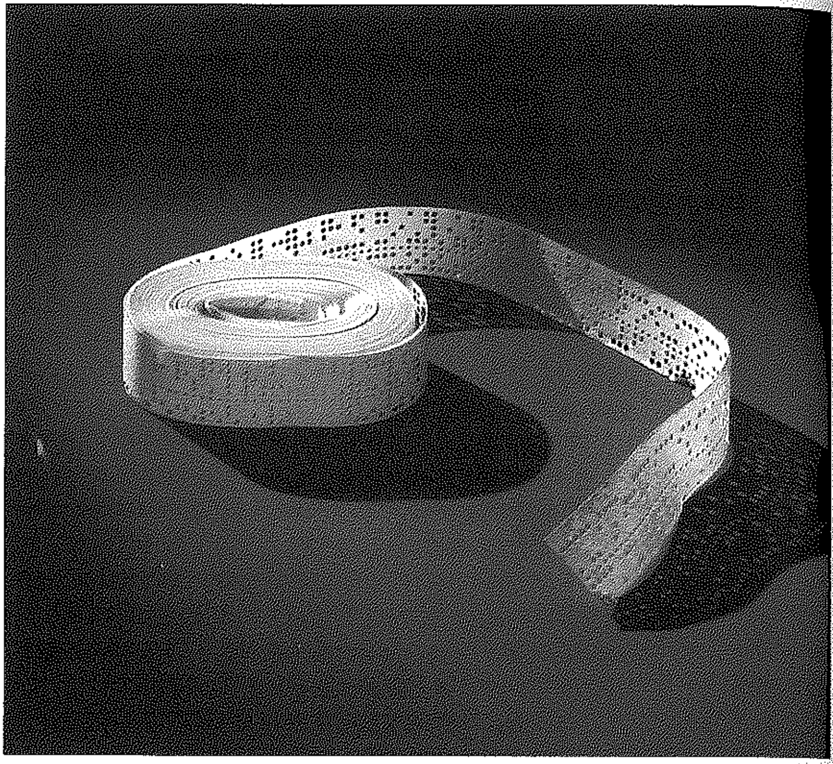

Figure 7.6
Paper tape containing BASIC, version 1.1, from the Smithsonian Collections. According to a letter by Bill Gates in the December 1975 issue of the Altair Users Group newsletter, *Computer Notes*: "If anyone is using BASIC version 1.1, you have a copy of a tape that was stolen back in March. No customers were ever shipped 1.1, as it was experimental and is full of bugs!" (*Source:* Smithsonian Institution.)

pass from BASIC to machine language easily—a crucial feature for getting a small system to do useful work.

These extensions kept their BASIC within its memory constraints while giving it the performance of a more sophisticated language. Yet it remained an interactive, conversational language that novices could learn and use. The BASIC they wrote for the Altair, with its skillful combination of features taken from Dartmouth and from the Digital Equipment Corporation, was the key to Gates's and Allen's success in establishing a personal computer software industry.

The developers of this language were not formally trained in computer science or mathematics as were Kemeny and Kurtz. They were introduced to computing in a somewhat different way. Bill Gates's private school in Seattle had a General Electric time-sharing system available for its pupils in 1968, a time when few students even in universities had such access. Later on he had access to an even better time-shared system: a PDP-10 owned by the Computer Center Corporation. Later still, he worked with a system of PDP-10s and PDP-11s used to control hydroelectric power for the Bonneville Power Administration. One of his mentors at Bonneville Power was John Norton, a TRW employee who had worked on the Apollo Program and who was a legend among programmers for the quality of his work.[91]

When he was writing BASIC for the Altair, Gates was at Harvard. He did not have access to an 8080-based system, but he did have access to a PDP-10 at Harvard's computing center (named after Howard Aiken). He and fellow student Monte Davidoff used the PDP-10 to write the language, based on the written specifications of the Intel 8080. In early 1975 Paul Allen flew to Albuquerque and demonstrated it to Roberts and Yates. It worked. Soon after, MITS advertised its availability for the Altair. Others were also writing BASIC interpreters for the Altair and for the other small computers now flooding the market, but none was as good as Gates's and Allen's, and it was not long before word of that got around.

It seemed that Roberts and his company had made one brilliant decision after another: the 8080 processor, the bus architecture, and now BASIC. However, by late 1975 Gates and Allen were not seeing it that way. Gates insists that he never became a MITS employee (although Allen was until 1976), and that under the name "Micro Soft," later "Micro-Soft," he and Allen retained the rights to their BASIC.[92] In a now-legendary "Open Letter to Hobbyists," distributed in early 1976, Gates complained about people making illicit copies of his BASIC by duplicating the paper tape. Gates claimed "the value of the computer time we have used [to develop the language] exceeds $40,000." He said that if he and his programmers were not paid, they would have little incentive to develop more software for personal computers, such as an APL language for the 8080 processor. He argued that illicit copying put all personal computing at risk: "Nothing would please me more than to hire ten programmers and deluge the hobby market with good software."[93]

Gates did his initial work on the PDP-10 while still an undergraduate at Harvard. Students were not to use that computer for commercial purposes, although these distinctions were not as clear then as they would be later. The language itself was the invention of Kemeney and Kurtz of Dartmouth; the extensions that were crucial to its success came from programmers at the Digital Equipment Corporation, especially Mark Bramhall, who led the effort to develop a time-sharing system (RSTS-11) for the PDP-11. Digital, the only commercial entity among the above group, did not think of its software as a commodity to sell; it was what the company did to get people to buy hardware.[94]

Bill Gates had recognized what Roberts and all the others had not: that with the advent of cheap, personal computers, software could and should come to the fore as the principal driving agent in computing. And only by charging money for it—even though it had originally been free—could that happen. By 1978 his company, now called "Microsoft," had severed its relationship with MITS and was moving from Albuquerque to the Seattle suburb of Bellvue. (MITS itself had lost its identity, having been bought by Pertec in 1977.) Computers were indeed coming to "the people," as Stewart Brand had predicted in 1972. But the driving force was not the counterculture vision of a Utopia of shared and free information; it was the force of the marketplace. Gates made good on his promise to "hire ten programmers and deluge the . . . market" (figure 7.7).

System Software: The Final Piece of the Puzzle

Gary Kildall's entree into personal computing software was as a consultant for Intel, where he developed languages for system development. While doing that he recognized that the floppy disk would make a good mass storage device for small systems, if it could be properly adapted. To do that he wrote a small program that managed the flow of information to and from a floppy disk drive. As with the selection of BASIC, it appears in hindsight to be obvious and inevitable that the floppy disk would be the personal computer's mass storage medium. That ignores the fact that it was never intended for that use. As with the adaptation of BASIC, the floppy had to be recast into a new role. As with BASIC, doing that took the work of a number of individuals, but the primary effort came from one man, Gary Kildall.

A disk had several advantages over magnetic or paper tape. For one, it was faster. For another, users could both read and write data on it. Its primary advantage was that a disk had "random" access: Users did not have to run through the entire spool of tape to get at a specific piece of data. To accomplish this, however, required tricky programming—some-

Figure 7.7
Microsoft Team, ca. 1978. This photograph shows Microsoft as it was moving from Albuquerque, where the Altair was built, to the Seattle area, where Bill Gates (lower left) and Paul Allen (lower right) were from. It was still a small company that focused mainly on supplying programming languages for personal computers. (*Source:* Microsoft.)

thing IBM had called, for one of its mainframe systems, a Disk Operating System, or DOS.[95]

A personal computer DOS had little to do with mainframe operating systems. There was no need to schedule and coordinate the jobs of many users: an Altair had one user. There was no need to "spool" or otherwise direct data to a roomful of chain printers, card punches, and tape drives: a personal computer had only a couple of ports to worry about. What *was* needed was rapid and accurate storage and retrieval of files from a floppy disk. A typical file would, in fact, be stored as a set of fragments, inserted at whatever free spaces were available on the disk. It was the job of the operating system to find those free spaces, put data there, retrieve it later on, and reassemble the fragments. All that gave the user an illusion that the disk was just like a traditional file cabinet filled with folders containing paper files.

Once again, Digital Equipment Corporation was the pioneer, in part because of its culture; because of the experience many of its employees had had with the TX-0 at MIT, one of the first computers to have a conversational, interactive feel to it. For its early systems DEC introduced DECtape, which although a tape, allowed programmers rapid access to data written in the middle, as well as at the ends, of the reel.[96] The PDP-10s had powerful DECtape as well as disk storage abilities; its operating systems were crucial in creating the illusion of personal computing that had so impressed observers like Stewart Brand.

In the late 1960s DEC produced OS/8 for the PDP-8, which had the feel of the PDP-10 but ran on a machine with very limited memory. OS-8 opened everyone's eyes at DEC; it showed that small computers could have capabilities as sophisticated as mainframes, without the bloat that characterized mainframe system software. Advanced versions of the PDP-11 had an operating system called RT-11 (offered in 1974), which was similar to OS/8, and which further refined the concept of managing data on disks.[97] These were the roots of personal computer operating systems. DEC's role in creating this software ranks with its invention of the minicomputer as major contributions to the creation of personal computing.

Gary Kildall developed PL/M for the Intel 8080. He used an IBM System/360, and PL/M was similar to IBM's PL/I. While working on that project Kildall wrote a small control program for the mainframe's disk drive. "It turned out that the operating system, which was called CP/M for Control Program for Micros, was useful, too, fortunately."[98] Kildall said that PL/M was "the base for CP/M," even though the commands were clearly derived from Digital's, not IBM's software.[99] For example, specifying the drive in use hy a letter; giving file names a period and three-character extension; and using the DIR (Directory) command, PIP, and DDT were DEC features carried over without change.[100] CP/M was announced to hobbyists as "similar to DECSYSTEM 10" in an article by Jim Warren in *Dr. Dobb's Journal of Computer Calisthenics and Orthodontia* [sic] in April 1976. Warren was excited by CP/M, stating that it was "well designed, based on an easy-to-use operating system that has been around for a DECade.[sic]"[101] Suggested prices were well under $100, with a complete floppy system that included a drive and a controller for around $800—not cheap, but clearly superior to the alternatives of cassette, paper tape, or any other form of tape. CP/M was the final piece of the puzzle that, when made available, made personal computers a practical reality.

Gary Kildall and his wife, Dorothy McEwen, eased themselves into the commercial software business while he also worked as an instructor at the Naval Postgraduate School in Monterey, California (figure 7.8). As interest in CP/M picked up, he found himself writing variations of it for other customers. The publicity in *Dr. Dobb's Journal* led to enough sales to convince him of the potential market for CP/M. In 1976 he quit his job and with Dorothy founded a company, Digital Research (initially Intergalactic Digital Research), whose main product was CP/M.[102]

The next year, 1977, he designed a version with an important difference. IMSAI, the company that had built a "clone" of the Altair (figure 7.9), wanted a license to use CP/M for its products. Working with IMSAI employee Glen Ewing, Kildall rewrote CP/M so that only a small portion of it needed to be customized for the specifics of the IMSAI. The rest would be common code that would not have to be rewritten each time a new computer or disk drive came along. He called the specialized code the BIOS—Basic Input/Output System.[103] This change standardized the system software in the same way that the 100-pin Altair bus had

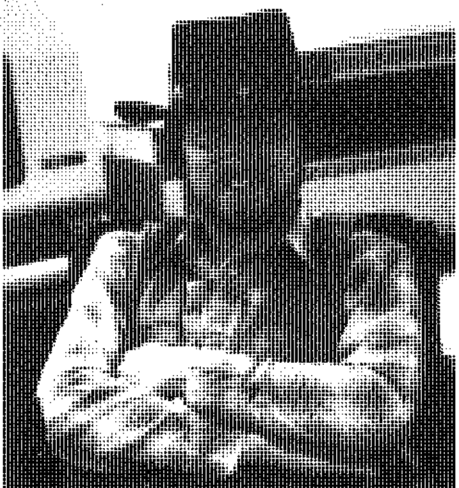

Figure 7.8
Gary Kildall. A DEC VT-100 terminal is visible in the background. (*Source:* Kristen Kildall.)

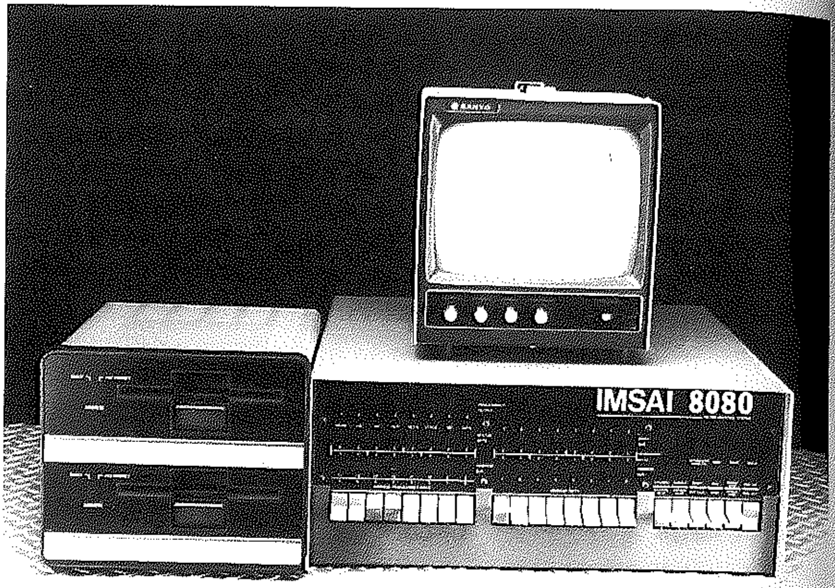

Figure 7.9
IMSAI 8080, one of the most successful copies of the Altair, with a video monitor and a disk storage system supplied by Micropolis. (*Source:* Smithsonian Institution.)

standardized hardware. IMSAI's computer system became a standard, with its rugged power supply, room for expansion with plenty of internal slots, external floppy drive, and CP/M.

End of the Pioneering Phase, 1977

By 1977 the pieces were all in place. The Altair's design shortcomings were corrected, if not by MITS then by other companies. Microsoft BASIC allowed programmers to write interesting and, for the first time, serious software for these machines. The ethic of charging money for this software gave an incentive to such programmers, although software piracy also became established. Computers were also being offered with BASIC supplied on a read-only-memory (ROM), the manufacturer paying Microsoft a simple royalty fee. (With the start-up codes also in ROM, there was no longer a need for the front panel, with its array of lights and switches.) Eight-inch floppy disk drives, controlled by CP/M, provided a way to develop and exchange software that was independent of particular models. Machines came with standardized serial and parallel ports, and connections for printers, keyboards, and video monitors. Finally, by 1977 there was a strong and healthy industry of publications, software companies, and support groups to bring the novice on board. The personal computer had arrived.

8
Augmenting Human Intellect, 1975–1985

In the mid-1970s, amid the grassroots energy and creativity of the small systems world, what else was happening? When the established computer companies saw personal computers appear, they, too, entered a period of creativity and technical advance. At first there was little overlap. By 1985, though, there was overlap and more: the paradigm of personal computing based on inexpensive microprocessors forced itself onto the industry. This chapter looks at how that happened.

Digital Equipment Corporation

Digital Equipment Corporation built the foundation for interactive personal computing with its minicomputers and its software. What were they doing when Intel announced its 8080, a device with the essentials of a minicomputer on one chip? "We were just in the throes of building the VAX."[1] The VAX was an extension of the PDP-11 that reached toward mainframe performance. It was a major undertaking for DEC and strained the company's resources. As IBM had done with its System/360, Digital "bet the company" on the VAX—a move toward higher performance and larger systems.

Many within DEC felt that the company was not so much a minicomputer builder as it was a company that sold architecture.[2] Beginning with the TX-0, DEC's founders had taken pride in their ability to build high-performance computers—large or small—through innovative design. That may explain why DEC failed to counter the threat that companies like Intel posed to its business. To build a computer around the Intel 8080 meant surrendering decisions about architecture to a semiconductor house—how could they allow themselves to do that? The other alternative, licensing the PDP-11 instruction set to chip makers, who would produce microprocessors based on it, was likewise rejected.

The company thought that would be giving the "corporate jewels" away. Digital did produce the LSI-11, a single-board PDP-11, in 1974, but that did not lead to inexpensive systems as did the Intel 8080. A single-chip PDP-11, called T-11, was developed but never marketed. The microprocessor phenomenon passed the PDP-11 by, even though elements of its architecture turned up in microprocessor designs (especially the Motorola 6800).[3]

Planning for an extension to the PDP-11 began in 1974 or 1975. DEC announced the VAX, Model 11/780, in October 1977 (figure 8.1). The full name was VAX-11, which stood for Virtual Address eXtension [of the] PDP-11. The implication was that the VAX was simply a PDP-11 with a 32-bit instead of a 16-bit address space. In fact, the VAX was really a new machine. It could, however, execute existing PDP-11 software by setting a "mode bit" that called forth the PDP-11 instruction set. (Eventually the compatibility mode was dropped.)

DEC continued to market small computers at successively lower prices and in smaller packages, for example, the PDP-8/A, introduced in 1975 for under $3,000.[4] But the company preferred to develop and market higher performance. One reason it gave was that for a given application, the cost of the computer was only part of the total cost; there was also

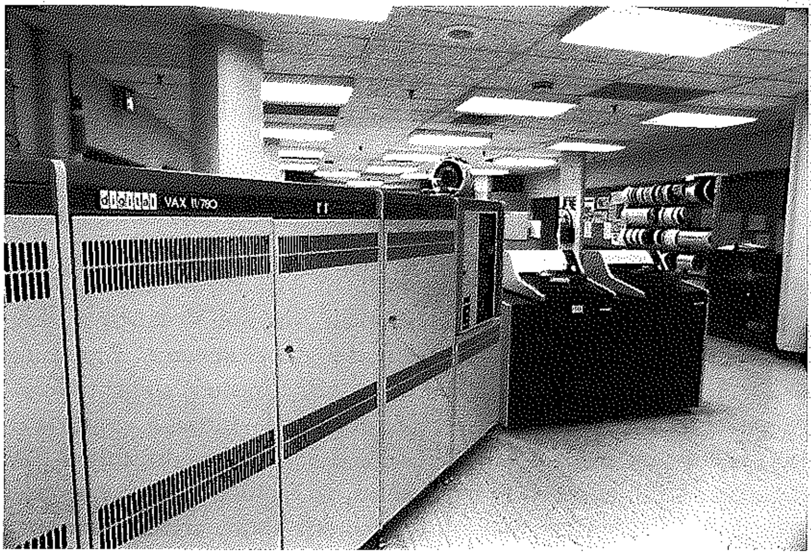

Figure 8.1
VAX 11/780, ca. 1978. (*Source:* Digital Equipment Corporation.)

"the high fixed overhead costs associated with the [existing] application."[5] Apparently DEC did not feel it could achieve truly drastic price reductions, as MITS had done with the Altair. That argument, coupled with DEC's reluctance to turn over its skill in computer architecture to semiconductor companies, kept the company out of the personal computer market during the crucial years, 1974 to 1977, when it could most easily have entered it.

Just as DEC was not the first to market a 16-bit mini, it was not the first to extend address space beyond 16 bits. In 1973, Prime, also located off Route 128 in Massachusetts, shipped a 32-bit minicomputer. Prime grew rapidly until merging with Computervision in the late 1980s. Another company, Interdata, described a "mega-mini" in 1974. Their design was also commercially successful, and that year the company was bought by Perkin-Elmer, the Connecticut optics company.[6] Systems Engineering Laboratory of Fort Lauderdale, Florida, also introduced a 32-bit mini, which was popular with NASA and aerospace customers. S.E.L. was sold to Gould in 1980 and became the basis for that venerable company's entree into the computer business.[7] The impetus for these developments was the growing availability of relatively cheap semiconductor memory to replace magnetic core. These memory chips made it more practical to design machines with large main memories, which in turn demanded more address space.

If the VAX was only nominally an extension of the PDP-11, it was genuinely a "virtual" memory computer. An informal definition of this term is that it is a way to make a computer's small but fast main memory seem to be bigger than it is, by swapping data to and from a slower but larger memory on a disk. A more precise definition concerns the way this is done: first of all, overall performance must not be seriously degraded by this process, and second, the user should not have to know that this swapping is going on (hence the term: the memory is "virtually" large but in reality small).[8]

The need for a hierarchy of memories, each slower but larger than the one below it, was discussed in the Institute for Advanced Study reports by Burks, Goldstine, and von Neumann in the late 1940s. The Atlas, designed at Manchester University in England and built by Ferranti in 1962, was probably the first to use a design that gave the user the illusion of a single-level fast memory of large capacity.[9] It was one of the fastest computers in the world at the time and also one of the most influential on successive generations. A user of the Atlas saw a machine with a virtual memory of one million 48-bit words. The computer automatically swapped data between the core and the drum, based on the contents of a set of registers (a technique called associative memory-addressing).[10] Though influential, commercial versions of the Atlas were only a modest success for Ferranti. In the United States, Burroughs offered virtual memory, with some important architectural advances, in the mid-1960s. IBM offered it with System/370 models announced in 1972. (It is probably from marketing the 370 that the term virtual memory came into wide use.)[11] The SDS 940 time-sharing system also followed the Atlas design.

C. Gordon Bell led the initial design effort for the VAX, and Bill Strecker was its chief architect. Breaking through the limits of the PDP-11's 65 Kbyte address space was their primary goal. The VAX provided 2^{32} or 4.3 gigabytes (equivalent to one billion 32-bit words) of virtual address space. Its addressing scheme divided memory into blocks, called pages, and used an associative comparison to determine whether the desired page was in core or not. The VAX processor used sixteen 32-bit general registers, like the IBM 360. It also had a rich set of over 250 instructions with nine different addressing modes, which allowed a single instruction to carry out complex operations.[12]

The VAX was a commercial success, selling around 100,000 over the next decade and leaping over the other 32-bit minis even though it appeared later. The 11/780's performance, roughly calculated at one million instructions per second (MIPS), became a benchmark against which competitors would compare their machines into the 1990s. A whole family of "Vaxen" followed: the less-powerful 11/750 in 1980, the higher-performance 8600 in 1984, and the compact MicroVax II in 1985, among others.[13] These machines kept DEC profitable and dominant along Route 128. Even Data General, whose Nova had been such a strong competitor for the PDP-11, had trouble competing with the VAX, although it did introduce a 32-bit Eclipse in 1980, as chronicled in Tracy Kidder's bestseller *The Soul of a New Machine.*[14]

The VAX was a general-purpose computer that came with the standard languages and software. It sold to a wide market, but its biggest impact was on engineering and science. Prices started at $120,000, which was too expensive for a single engineer, but just cheap enough to serve a division at an aerospace, automotive, or chemical firm. For them the standard practice had been either to get in line to use the company's mainframe, or to sign up for time on a commercial time-sharing service. The VAX gave them computing power at hand. It had a solid, engineering-oriented operating system (VMS), and sophisticated I/O facilities for data collection.

Finally, the VAX came with a powerful and easy-to-use terminal, the VT-100. It had an impressive number of features, yet one felt that none was superfluous. It somehow managed to retain the comfortable feel of the old Teletype. One feature that many users loved was its ability to scroll a pixel at a time, rather than a line at a time. There was no practical reason to have this feature, and it failed to catch on with other terminal displays, but it had a great appeal. The VT-100's codes, using ASCII, did become the standard for terminals for the next twenty years.

A Word about UNIX

The impact of UNIX on commercial computing will be discussed more fully in the next chapter, and here we will just briefly describe its place with regard to the VAX. In addition to VMS, the VAX's PDP-11 ancestry meant that users could also run UNIX on it. UNIX was developed on DEC minicomputers, and for the first few years of UNIX's existence it ran only on DEC computers, mainly PDP-11s. The University of California at Berkeley's version of UNIX, which had an enormous influence on computing and on the Internet, was developed on a VAX. Still, DEC was ambivalent about UNIX for the VAX. Ken Olsen allegedly stated at one point that "UNIX is snake oil!" (The context in which he made that statement has been disputed.[15]) At any rate, the VAX could and did run Berkeley UNIX, and for at least the formative years, VAX computers were the most common nodes on the Internet.[16]

IBM and the Classic Mainframe Culture

In the mid-1970s, while the personal computer was being invented and while Digital was building the VAX, what was IBM doing? Like Digital, IBM was busy extending its existing line, with the high-end 3033 announced in early 1977, and the low-priced 4300-series announced in 1979. This latter series offered a dramatic increase in performance per dollar over the mid-range 370 systems then being marketed, an improvement that came mainly from using large-scale integrated circuits. These LSI chips were developed and designed by IBM in-house and did not resemble the ones being marketed by companies like Intel or Fairchild.[17] As System/370 installations grew in number and complexity, the issue of interconnecting them also arose. Bob Evans of IBM remarked that, in the early 1970s, the plethora of incompatible and ad hoc networking schemes resembled the chaos of computer architectures that IBM had sought to reduce a decade before.[18] The result was Systems

Network Architecture (SNA), first shipped in 1974. SNA was a layer cake of standards, spelled out in detail. It formed the basis for networking large computer systems into the 1990s.

In 1975 IBM introduced a product that might have seemed at odds with its mainframe orientation: a "personal" computer, Model 5100. This machine could fit on a desk and contained a processor, keyboard, cassette tape drive, and small video terminal in a single package. Prices began at $9,000 for a machine with 16 Kbytes of memory.[19] It supported both BASIC and APL (the developer of APL, Kenneth Iverson, had joined IBM in 1960), which the user could select by flipping a switch on the front panel. But little or no applications software was available; the third-party support community that grew up around the Altair failed to materialize for the 5100. Sales were modest but steady. (The "other" IBM personal computer will be discussed shortly.)

Another answer to the question of what IBM was doing is that it was in court. For IBM the 1970s was the decade of the lawsuit: *U.S. vs. IBM*, filed January 17, 1969, and dismissed in 1981. The charge was that IBM was in violation of antitrust laws by virtue of its dominance of the U.S. market for general-purpose electronic digital computers. The Justice Department based this charge on a definition of "market" that covered the business-oriented electronic data-processing activities served by mainframe computers, of which IBM held about 70 percent of the market and the "BUNCH" nearly all the rest. IBM countered by arguing that its competition was not just Burroughs, Univac, NCR, CDC, and Honeywell, but rather thousands of companies, large and small, that made and sold computers, peripherals, software, services, and the like. After a long discovery process, during which depositions were taken from representatives of most of these companies, the case finally went to trial in May, 1975—that is, around the time that Bill Gates and Paul Allen were talking about developing BASIC for the Altair.

The discovery process and the testimony were thorough and detailed. Transcripts of the depositions and testimony run into thousands of pages.[20] But none of the gathering storm of personal systems made it into the trial. Neither Bill Gates nor Ed Roberts was called to testify or give depositions. The court focused its attention on the former "Dwarfs," especially RCA and GE, who had left the business. Occasionally firms that competed with IBM's mainframes at one or two places were examined. These included SDS (a subsidiary of Xerox by then), whose computers competed with the System/360 Model 44 for scientific applications, and Digital Equipment Corporation, not for its minicomputers but for its PDP-10. The court even looked closely at Singer, the venerable sewing machine company, which had purchased Friden in 1963 and built up a business in point-of-sale retail terminals. (The British company ICL bought Singer's computing business in 1976.)

Reading through the volumes of transcripts, one feels a sense of tragedy and unreality about the whole trial. The judge, David Edelstein, was often baffled by the avalanche of jargon that spewed forth from the expert witnesses each day; this typically resulted in his losing his temper by mid-afternoon. (The courtroom had a defective air-conditioning system, which did not help matters in the summer, either.) The money spent on hiring and retaining a team of top-notch attorneys (led by Nicholas Katzenbach for IBM) and their research staffs was money that did not go into the research and development of new computer technology. And yet both sides, with all their highly paid legal and research staffs, utterly and completely missed what everyone has since recognized as the obvious way that computing would evolve: toward microprocessor-based, networked desktop computing. There is no record of someone bringing an Apple II into the courthouse building in lower Manhattan; if someone had, would anyone have recognized it for what it was? By coincidence, just as the Apple II was being introduced at a computer fair in California in 1977, one expert witness testified, "I will be a little stronger than that . . . it is most unlikely that any major new venture into the general purpose [sic] computer industry can be expected."[21] As late as 1986 one Justice Department economist, still fuming over the dismissal of the case, complained that "IBM faces no significant domestic or foreign competition that could threaten its dominance."[22] That statement was made the year that Microsoft offered its shares to the public. A few years later IBM began suffering unprecedented losses and began laying off employees for the first time in its history. A new crop of books soon appeared, these telling the story of how IBM had been outsmarted by Bill Gates. Other than writing tell-all books about IBM, everything else about the computer industry had fundamentally and irrevocably changed.[23]

In the end the combatants ran out of energy. The 1981 inauguration of Ronald Reagan, who had campaigned against an excessive exercise of federal power, was enough to end it. But what really killed the government's case was that, even neglecting the personal computer, there was vigorous and healthy competition throughout the decade. The failures of GE and RCA were more than offset by the successes of Digital Equipment, SDS, Amdahl, and software companies like EDS. The industry was too healthy: the personal fortunes amassed by Gene Amdahl and Max Palevsky made it hard to take the charges seriously.[24] In one of the rare instances of levity, IBM's lawyers were able to elicit more than a few chuckles in the courtroom when they described the enormous wealth that Palevsky—a philosophy student—made in a few years with machines aimed right at IBM's middle range of mainframes.

IBM continued to develop new products. In addition to the 4300 and 3030 mainframes, IBM went after the minicomputer companies with its System/38 in 1978, following that with its AS/400 in 1988. The AS/400 was aimed more at business than engineering customers, but otherwise it was a strong competitor to the VAX. It used advanced architectural features that IBM had planned for its follow-on to the System/370 but had not implemented. As such, the AS/400 represented IBM's most advanced technology, and it generated strong revenues for IBM into the 1990s, when its mainframe sales suffered from technological obsolescence.[25] IBM failed to bring other products to market at this time, however, a failure that ultimately hurt the company. It is not clear how much the antitrust suit had to do with that.

From "POTS" to "OLTP"

The concept of a computer utility, naively envisioned in the late 1960s as being like the electric power utilities, evolved in several directions in the 1970s. General Electric built a large international network from its association with Dartmouth. Using machines like the PDP-10 and SDS 940, other utilities offered unstructured computer time. By 1975 TYMNET comprised a network of twenty-six host computers, routed through eighty communications processors. The simple hub-and-spoke topology of time-sharing evolved into a web of multiple rings, so that the failure of one host would not bring the system down.[26]

At the same time, a more tightly structured and disciplined use of terminals for on-line access also appeared. This was tailored for a specific application, such as processing insurance claims, where the programs and types of data handled by a terminal were restricted. Many were private, though some were semipublic, such as the effort by the U.S. National Library of Medicine to put its century-old *Index Medicus* on-line. (By the end of the 1970s its *MEDLINE* system provided on-line searches of medical literature from research libraries worldwide.) These systems were more like the SAGE air-defense and SABRE airline reservations systems of the late 1950s than they were like the Dartmouth College model. A new acronym appeared to describe it, "OLTP" for "On-line Transaction Processing," to differentiate it from the less-structured connotations of "POTS" (Plain Old Time-Sharing). Thus although computer usage was no longer in batches of cards, some of the basic structure of a punched-card installation remained.

A number of companies introduced terminals to serve this market. Some were descended from the Teletype and combined a typewriter keyboard and a printing mechanism (e.g., the DECwriter II or Teletype Model 37, both ca. 1975). Others replicated the Teletype, only with a video screen instead of a printer. These, like the Lear-Siegler ADM-3, were sometimes called "dumb terminals," "glass teletypes," or "glass TTY": they offered little beyond simple data entry and viewing. In contrast to them were "smart" terminals that allowed users to see and edit a full screen of text, and which contained a measure of computing power. Besides the VT-100, DEC had produced several designed around a PDP-8 processor; another company that had some success was Datapoint of San Antonio. Recall that it was Datapoint's contract with Intel that led to the 8080 microprocessor; however, the Datapoint 2200 terminal did not use a microprocessor. Some of these terminals, especially the Datapoint, came close to becoming personal computers without the vendor realizing it.[27]

The VT-100 became the standard ASCII terminal, while a terminal introduced by IBM became the EBCDIC standard by 1980. That was the model 3270, announced in 1971.[28] The 3270 was the philosophical opposite of the DEC VT-100: it operated on the assumption that the user would be keying structured information into selected fields, with other fields (e.g., for a person's name or date of birth) replicated over and over for each record. Therefore, the terminal did not transmit information as it was keyed in but waited until a full screen was keyed in; then it sent only whatever was new to the computer (in compressed form). IBM mainframe installations now routinely included terminals and time-shared access through the time sharing option (TSO) software. Typically these terminals were segregated in special rooms near the mainframe installation. They were seldom found in a private office.

By 1980, as the lawsuit was coming to an end, IBM still dominated the industry. But more and more, IBM was floating in a slower channel of the river. That began in 1963 with the development of ASCII, when IBM adopted EBCDIC. In 1964 IBM chose a hybrid semiconductor technology over ICs. In 1970 it adopted integrated circuits of its own design, slightly different from the standard TTL chips then flooding the market.

In the mid-70s, IBM's Systems Network Architecture established a standard for networking large systems, but SNA was different from the networking schemes then being developed by the Defense Department. Finally, there were the different approaches to terminal design represented by the 3270 and VT-100. Only with hindsight can we discern a pattern.

IBM's introduction of the personal computer in 1981 brought the issue to a head. The IBM PC used ASCII, not EBCDIC. It used standard TTL and MOS chips from outside suppliers. And its connections with its keyboard and monitor were closer to the minicomputer than to the 3270. The PC's internal design reveals how the pressures of the marketplace were able to accomplish what the courts and the U.S. Justice Department could not do.

The mainframe, batch model of computing, whose high-water mark was the 7090, was giving way, not only to interactive but also to decentralized processing. The increasingly fuzzy line that distinguished "smart" terminals from stand-alone personal computers was one indication. New design questions came to the fore: how to apportion processing functions and memory among the terminals and a central system, and how to send data efficiently and reliably through such a network. IBM had embarked on the design of a "Future System" (FS) that attacked some of these issues head-on. Planning for FS began in the early 1970s, and IBM hoped to announce products to replace its System/370 line by 1975. But FS was abandoned in 1975, in part because its designers were unable to solve the architectural problems, and in part because the success of the System/370 architecture meant that IBM would put itself at an unacceptable risk to abandon that market to third-party vendors.[29] Some of the concepts found their way into the mid-range AS/400, but canceling FS was "the most expensive development-effort failure in IBM's history."[30]

Viatron

A start-up company from Route 128 had an idea with similar promise but equally dismal results. The John the Baptist of distributed computing was Viatron Computer Systems of Bedford, Massachusetts. It was the outgrowth of an Air Force Project from the mid-1960s called AESOP (Advanced Experimental System for On-line Planning). Prepared by the MITRE Corporation, AESOP envisioned a network of terminals that provided visual as well as text information to middle and high-level managers, including those without any sophistication in computing, to help them do their work with the same level of acceptance as the telephone:

The core of the management system . . . will be not so much the central processor or central memory. The real basis . . . will be the unique program of instructions which makes the central processor, the central memory, and the organization's store of data and formal quantitative models easily available to the manager through the window of his desk top display, thus making it possible for him to exert the full power of his intentions through the use of his simple lightgun pointer. As AESOP-type management systems are developed, managers will learn to converse and interact with the processor with ease and naturalness. They will also learn to communicate through the processor with other members of the organization.[31]

Two of the report's authors, Joseph Spiegel and Dr. Edward Bennett, left MITRE and cofounded Viatron in 1967. Bennett was successful in raising venture capital—these were the go-go years—and announced that by 1969 Viatron would be renting interactive terminals that would move processing onto the desktop. He also predicted that his company would surpass IBM in numbers of installed computers. System 21 terminals were to rent for the unbelievably low price of $40 a month.[32] The system included a keyboard, a 9-inch video display, and two cassette tape drives for storage of data and formatting information (figure 8.2). An optional attachment allowed users to disconnect the keyboard and tape unit and connect it to any standard television set for remote computing, say, in a hotel room. The terminal contained within it a "micro-processor" [sic] with 512 characters of memory. Other options included an optical character-recognition device, a "communications adapter," and an ingenious, Rube Goldberg–inspired "printing robot" which one placed over a standard IBM Selectric typewriter. Activating a set of solenoids, mechanical fingers pressed the Selectric keys to type clean output at 12 characters/second.[33]

The key to Viatron's impressive specifications was its use of MOS integrated circuits. This technique of integrated-circuit fabrication was the technical foundation for the microprocessor revolution of the 1970s, but it was immature in 1969. Viatron had to invest its start-up capital in perfecting MOS, and then it needed more money to gear up for volume production. That was too ambitious. By 1970, production lines were just starting, but the volume was small, and Viatron's sales and marketing were in disarray. At a meeting of the board held in Bennett's home in the summer of 1970, he found himself ousted from Viatron just as his wife was about to serve everyone dinner (they never ate the meal). The

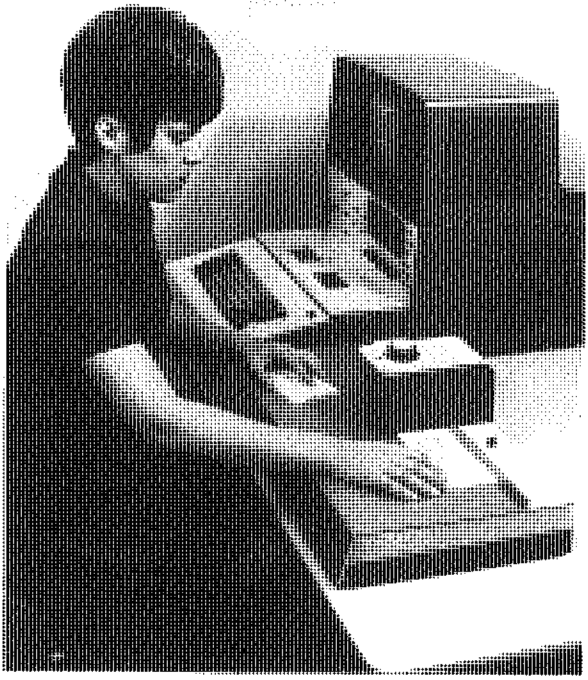

Figure 8.2
Office automation: Viatron 21. (*Source:* Charles Babbage Institute, University of Minnesota.)

company delivered a few systems by 1971, but in April of that year it declared bankruptcy. Losses to venture capitalists ran upwards of $30 million in fiscal year 1969–1970 alone.[34] Viatron became just another of many companies to fail while attempting to topple IBM from the top of the industry.

Wang

Even by the metric of the go-go years, Viatron's trajectory was bizarre, which should not obscure the truth of Bennett's observation. Advances in MOS integrated circuits were making IBM's way of doing computing obsolete, at least in the office environment. The company that succeeded where Viatron failed was Wang Labs, which in an earlier era had pioneered in electronic calculators. By 1971 Wang recognized that calculators were becoming a commodity, with razor-thin profit margins dependent on packaging more than on technical innovation. Wang Labs began a transition to a minicomputer company, and by 1972 made a complete crossover with its Model 2200 "computing calculator"—a general-purpose computer, although Wang was careful not to market it as such. Like Digital Equipment in the late 1950s, Wang was reluctant to use the word "computer" because of the word's connotations. Wang had an astute sense of knowing when to get out of one market and into a new one about to open up. As Wang's profits soared, Wall Street analysts concocted elaborate theories to explain this, some based on an alleged innate sense that his Chinese ancestors gave him. Dr. Wang was, in fact, a conservative engineer who understood the technology of his company's products and who valued his company's independence. In this regard he was closer to his Yankee counterpart and neighbor, Ken Olsen of DEC, than he was to any Western stereotype of Oriental mind-set.[35]

An Wang chose next to direct the company toward what would later be known as "Office Automation." In the mid-1970s that meant word processing. Word processing has become so commonplace that it is hard to recall how absurd the concept was at a time when even small computers cost thousands and skilled typists were paid $1.25 an hour. An old story tells of how graduate students at MIT programmed the $120,000 PDP-1 to serve as an "expensive typewriter" in the early 1960s. IBM developed a program called TEXT-90 for the 7090, but that was used only for special applications and never penetrated the office environment. In the early 1960s, some members of the committee working on the ASCII standard argued that codes for lowercase letters were unnecessary and a waste of space.[36]

The term "word processing" came into use after 1964, when IBM announced the MTST—a version of its Selectric typewriter that could store and recall sequences of keystrokes on a magnetic tape cartridge.[37] An early Wang product, the Model 1200, was similar, but customers found its complexity daunting. Other companies that entered the field at this time included NBI ("Nothing But Initials") outside of Denver, Lanier in Atlanta, and CPT in Minneapolis.

The second time around Wang got it right. Wang engineers found out first of all what office people wanted. They realized that many users of word-processing equipment were terrified of losing a day's work by the inadvertent pressing of the wrong key. And it wasn't just secretaries who were prone to such actions: in 1981 ex-President Jimmy Carter lost a few pages of his memoirs—"I had labored over them for a couple of days"—

by pressing the wrong key on his $12,000 Lanier "No Problem" [sic] word-processing system. An anxious phone call to Lanier produced a utilities disk that allowed him to recover the data from the original diskette.[38] After this, Wang's engineers came up with a design that would make such a loss nearly impossible. They also decided on a terminal that used a cathode ray tube, which displayed half a page of text instead of the one or few lines that other systems used. Commands were accessed by a simple screen of menus. In a later era Wang's design might have been known by the cliché "user-friendly"; it was also a "distributed" system. But the company used neither term in its marketing. Unlike other minicomputer companies, Wang did little OEM business; it sold machines to the people who were going to use it. Wang spared its customers—Wall Street brokerage houses, large banks, and oil companies at first—the technical jargon. (A decade later office workers were not so lucky, everything would get plastered with the term "user friendly" no matter how obtuse it was).[39]

A major requirement was that the system have a speedy response. Time-sharing relieved users of the need to wait in a queue with a deck of punched cards, but on a busy day users faced an equally onerous wait at their terminals while the mainframe got around to each job. Unlike MIT hackers, office employees could not be expected to come in at midnight to do their work. The answer was to put some of the processing power into the terminal itself, with the central computer serving primarily for data storage and retrieval—commonplace after 1985, but a radical departure from time-sharing in 1975. The WPS (Wang Word Processing System) was unveiled at a trade show in New York in June 1976, and according to some accounts nearly caused a riot (figure 8.3).[40] A basic system, including hard disk storage, cost $30,000. Wang Labs, ranked forty-fifth in data-processing revenues in 1976, moved up to eighth place by 1983, just below IBM, DEC, and the remnants of the BUNCH. Some analysts thought Wang was in the best position of any company to become number two in the industry. (No Wall Street person would risk his career by predicting a new number one.) Others put the company's success into the pigeonhole of "office automation" rather than general-purpose computing, but what Wang was selling was at heart a general-purpose, distributed computer system. Wang's success was a vindication of Viatron's vision. However, Wang was unable to reinvent itself once again in the 1990s, when it faced competition from commodity personal computers running cheap word-processing software, and it too went bankrupt.[41]

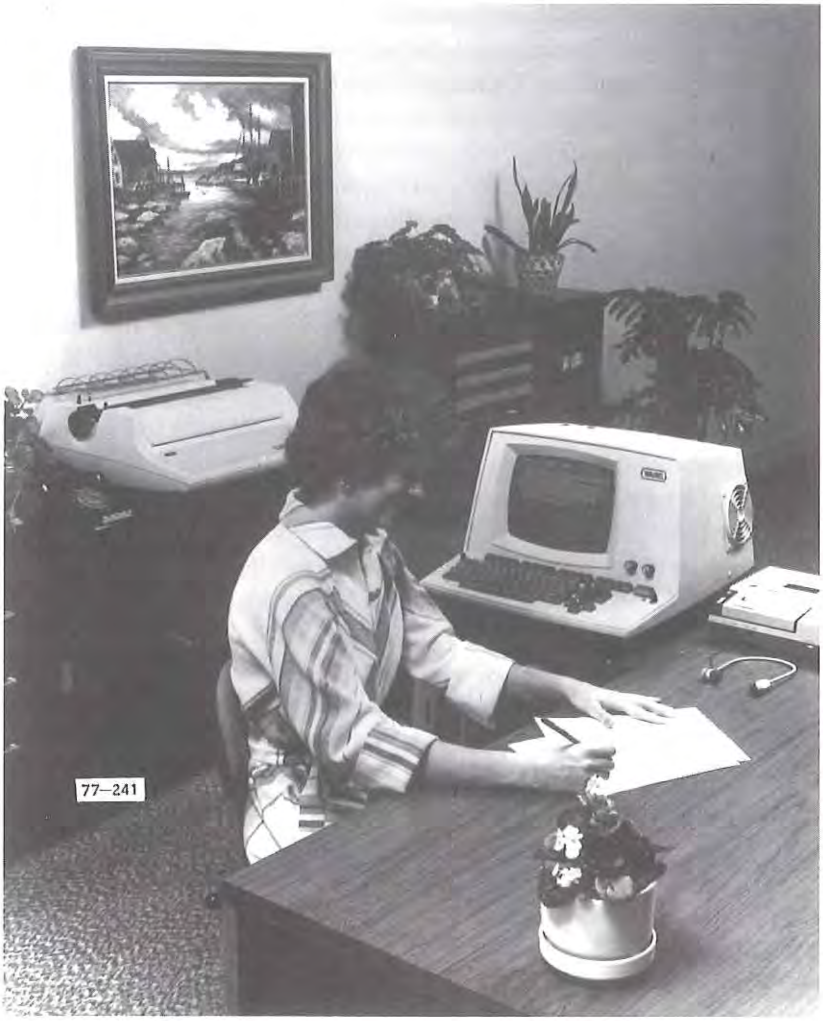

Figure 8.3
Office automation: WANG Word Processing System. (*Source:* Charles Babbage Institute, University of Minnesota.)

Xerox PARC

One of the ironies of the story of Wang is that despite its innovations, few stories written about the 1970s talk about Wang. To read the literature on these subjects, one would conclude that the Xerox Corporation was the true pioneer in distributed, user-friendly computing; that the Xerox Palo Alto Research Center, which Stewart Brand so glowingly described in his 1972 *Rolling Stone* article, was the place where the future of computing was invented. Why was that so?

The Xerox Corporation set up a research laboratory in the Palo Alto foothills in 1970. Its goal was to anticipate the profound changes that technology would bring to the handling of information in the business world. As a company famous for its copiers, Xerox was understandably nervous about talk of a "paperless office." Xerox did not know if that would in fact happen, but it hoped that its Palo Alto Research Center (PARC) would help the company prosper through the storms.[42]

Two things made PARC's founding significant for computing. The first was the choice of Palo Alto: Jacob Goldman, director of corporate research at Xerox, had favored New Haven, Connecticut, but the person he hired to set up the lab, George Pake, favored Palo Alto and prevailed, even though it was far from Xerox's upstate New York base of operations and its Connecticut headquarters. The lab opened just as "Silicon Valley," led by Robert Noyce of the newly founded Intel, was taking form.

The second reason for PARC's significance took place in the halls of Congress. As protests mounted on college campuses over the U.S. involvement in Viet Nam, a parallel debate raged in Congress that included the role of universities as places where war-related research was being funded by the Defense Department. Senator J. William Fulbright was especially critical of the way he felt science research was losing its independence in the face of the "monolith" of the "military-industrial complex" (a term coined by President Eisenhower in 1961). In an amendment to the 1970 Military Procurement Anthorization Bill, a committee chaired by Senator Mike Mansfield inserted langnage that "none of the funds authorized . . . may be used to carry out any research project or study unless such a study has a direct and apparent relationship to a specific military function or operation."[43] The committee did not intend to cripple basic research at universities, only to separate basic from applied research. Some members assumed that the National Science Foundation would take the DoD's place in funding basic research. Even before the passage of this "Mansfield Amendment," the DoD had moved to reduce spending on research not related to specific weapons systems; thus this movement had support among hawks as well as doves.

The NSF was never given the resources to take up the slack. At a few select universities, those doing advanced basic research on computing felt that they were at risk, because their work was almost entirely funded by the Defense Department's Advanced Research Projects Agency (ARPA).[44] At that precise moment, George Pake was scouring the country's universities for people to staff Xerox PARC. He found a crop of talented and ambitious people willing to move to Palo Alto. ARPA funding had not been indiscriminate but was heavily concentrated at a few universities—MIT, Carnegie-Mellon, Stanford, UC-Berkeley, UCLA, and the University of Utah—and researchers from nearly every one of them ended up at PARC, including Alan Kay and Robert Taylor from Utah, and Jerome Elkind and Robert Metcalfe from MIT.[45] There were also key transfers from other corporations, in particular from the Berkeley Computer Corporation (BCC), a struggling time-sharing company that was an outgrowth of an ARPA-funded project to adapt an SDS computer for time-sharing. Chuck Thacker and Butler Lampson were among the Berkeley Computer alumni who moved to PARC. All those cited above had had ARPA funding at some point in their careers, and Taylor had been head of ARPA's Information Processing Techniques Office.

Two ARPA researchers who did not move to PARC were the inspiration for what would transpire at Xerox's new lab. They were J.C.R. Licklider, a psychologist who initiated ARPA's foray into advanced computer research beginning in 1962, and Douglas Engelbart, an electrical engineer who had been at the Stanford Research Institute and then moved to Tymshare. In 1960, while employed at the Cambridge firm Bolt Beranek and Newman, Licklider published a paper titled "Man-Computer Symbiosis" in which he forecast a future of computing that "will involve a very close coupling between the human and electronic members of the partnership." In a following paper, "The Computer as a Communication Device," he spelled out his plan in detail.[46] He was writing at the heyday of batch processing, but in his paper Licklider identified several technical hurdles that he felt would be overcome. Some involved hardware limits, which existing trends in computer circuits would soon overcome. He argued that it was critical to develop efficient time-sharing operations. Other hurdles were more refractory: redefining the notions of programming and data storage as they were then practiced. In 1962 "Lick" joined ARPA, where he was given control over a fund that he could use to realize this vision of creating a "mechanically extended man."[47]

Douglas Engelbart was one of the first persons to apply for funding from ARPA's Information Processing Techniques Office in late 1962; he was seeking support for a "conceptual framework" for "augmenting human intellect."[48] Engelbart says that a chance encounter with Vannevar Bush's *Atlantic Monthly* article "As We May Think" (published in July 1945) inspired him to work on such a plan. Licklider directed him to work with the time-shared Q-32 experimental computer located in Santa Monica, through a leased line to Stanford; later Engelbart's group used a CDC 160A, the proto-minicomputer. The group spent its time studying and experimenting with ways to improve communication between human beings and computers. His most famous invention, first described in 1967, was the "mouse," which exhaustive tests showed was more efficient and effective than the light pen (used in the SAGE), the joystick, or other input devices.[49] Engelbart recalled that he was inspired hy a device called a planimeter, which an engineer slid over a graph to calculate the area under a curve. Among many engineers this compact device was a common as a slide rule; it is now found only among antique dealers and museums.

In December 1968 Engelbart and a crew of over a dozen helpers (among them Stewart Brand) staged an ambitious presentation of his "Augmented Knowledge Workshop" at the Fall Joint Computer Conference in San Francisco. Interactive computer programs, controlled by a mouse, were presented to the audience through a system of projected video screens and a computer link to Palo Alto. Amazingly, everything worked. Although Engelbart stated later that he was disappointed in the audience's response, the presentation has since become legendary in the annals of interactive computing. Engelbart did not join Xerox-PARC, but many of his coworkers, including Bill English (who did the detail design of the mouse), did.[50]

What was so special about the mouse? The mouse provided a practical and superior method of interacting with a computer that did not strain a user's symbolic reasoning abilities. From the earliest days of the machine's existence, the difficulties of programming it were recognized. Most people can learn how to drive a car—a complex device and lethal if not used properly—with only minimal instruction and infrequent reference to an owner's manual tossed into the glove box. An automobile's control system presents its driver with a clear, direct connection between turning the steering wheel and changing direction, pressing on the gas pedal and accelerating, pressing on the brake pedal and slowing down. Compare that to, say, UNIX, with its two- or three-letter commands, in which the command to delete a file might differ from one to print a file only by adjacent keys. Automobiles—and the mouse—use eye-hand coordination, a skill human beings have learned over thousands of years of evolution, but a keyboard uses a mode of human thought that humans acquired comparatively recently. Researchers at PARC refined the mouse and integrated it into a system of visual displays and iconic symbols (another underutilized dimension of human cognition) on a video screen.

For the U.S. computing industry, the shift of research from ARPA to Xerox was a good thing; it forced the parameters of cost and marketing onto their products. It is said that Xerox failed to make the transition to commercial products successfully; it "fumbled the future," as one writer described it. Apple, not Xerox, brought the concept of windows, icons, a mouse, and pull-down menus (the WIMP interface) to a mass market, with its Macintosh in 1984. Xerox invented a networking scheme called Ethernet and brought it to market in 1980 (in a joint effort with Digital and Intel), but it remained for smaller companies like 3-Com to commercialize Ethernet broadly. Hewlett-Packard commercialized the laser printer, another Xerox-PARC innovation. And so on.[51]

This critique of Xerox is valid but does not diminish the magnitude of what it accomplished in the 1970s. One may compare Xerox to its more nimble Silicon Valley competitors, but out of fairness one should also compare Xerox to IBM, Digital, and the other established computer companies. Most of them were in a position to dominate computing: DEC with its minicomputers and interactive operating systems, Data General with its elegant Nova architecture, Honeywell with its Multics time-sharing system, Control Data with its Plato interactive system, and IBM for the technical innovations that its research labs generated. Although they did not reap the rewards they had hoped for, each of these companies built the foundation for computing after 1980.

Within Xerox-PARC, researchers designed and built a computer, the Alto, in 1973 (figure 8.4). An architectural feature borrowed from the MIT-Lincoln Labs TX-2 gave the Alto the power to drive a sophisticated screen and I/O facilities without seriously degrading the processor's performance. Eventually over a thousand were built, and nearly all were used within the company. Networking was optional, but once available, few Alto users did without an Ethernet connection. An Alto cost about $18,000 to build. By virtue of its features, many claimed that the Alto was the first true personal computer. It was not marketed to the public, however—it would have cost too much for personal use.[52] Besides using

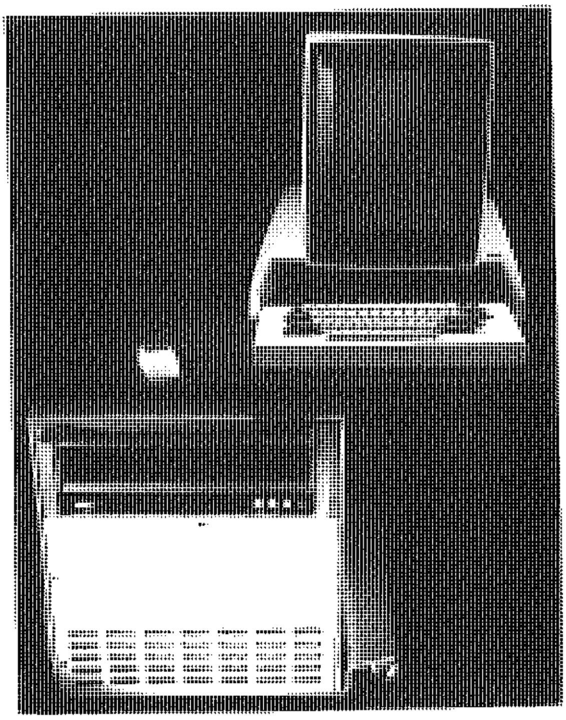

Figure 8.4
Xerox Alto, ca. 1973. (*Source:* Smithsonian Institution.)

a mouse and windows, the Alto also had a "bit-mapped" screen, where each picture element on the screen could be manipulated by setting bits in the Alto's memory. That allowed users to scale letters and mix text and graphics on the screen. It also meant that a text-editing system would have the feature "what you see is what you get" (WYSIWYG)—a phrase made popular by the comedian Flip Wilson on the television program "Laugh-In."[53]

In 1981 Xerox introduced a commercial version, called the 8010 Star Information System, announced with great fanfare at the National Computer Conference in Chicago that summer. Advertisements described an office environment that would be commonplace ten years later, even more capable than what office workers in 1991 had. But the product fizzled. Around the same time Xerox introduced an "ordinary" personal computer using CP/M, but that, too, failed to sell.[54]

The Star, derived from the Alto, was technically superior to almost any other office machine then in existence, including the Wang WPS. Personal computers would have some of the Star's features by 1984, but integrated networks of personal computers would not become common for another ten years. In the late 1970s, Wang had a better sense than Xerox of what an office environment was like and what its needs were. Advertisements for the Star depicted an executive calling up, composing, and sending documents at his desk; somehow Xerox forgot that business executives do not even place their own telephone calls but get a secretary to do that. By contrast, Wang aimed its products at the office workers who actually did the typing and filing. The Alto was more advanced, which explains why its features became common in office computing in the 1990s. The Wang was more practical but less on the cutting edge, which explains both Wang's stunning financial success in the late 1970s, and its slide into bankruptcy afterward.

Along with its invention of a windows-based interface, Xerox's invention of Ethernet would have other far-reaching consequences. Ethernet provided an effective way of linking computers to one another in a local environment. Although the first decade of personal computing emphasized the use of computers as autonomous, separate devices, by the mid-1980s it became common to link them in offices by some form of Ethernet-based scheme. Such a network was, finally, a way of circumventing Grosch's Law, which implied that a large and expensive computer would outperform a cluster of small machines purchased for the same amount of money. That law had held up throughout the turmoil of the minicomputer and the PC; but the effectiveness of Ethernet finally brought it, and the mainframe culture it supported, down.[55] How that happened will be discussed in the next chapter.

Personal Computers: the Second Wave, 1977–1985

Once again, these top-down innovations from large, established firms were matched by an equally brisk pace of innovation from the bottom up—from personal computer makers.

In the summer of 1977 Radio Shack began offering its TRS-80 in its stores, at prices starting at $400. The Model 1 used the Z-80 chip; it was more advanced than the Intel 8080 (although it did not copy the Altair architecture). The Model 1 included a keyboard and a monitor, and cassettes to be used for storage. A start-up routine and BASIC (not Microsoft's) were in a read-only memory. The marketing clout of Radio

Shack, with its stores all over the country, helped make it an instant hit for the company.[56] Because Radio Shack's customers included people who were not electronics hobbyists or hackers, the Model 1 allowed the personal computer to find a mass audience. Years later one could find TRS-80 computers doing the accounting and inventory of small businesses, for example, using simple BASIC programs loaded from cassettes or a floppy disk. The TRS-80 signaled the end of the experimental phase of personal computing and the beginning of its mature phase.

Two other computers introduced that year completed this transition. The Commodore PET also came complete with monitor, keyboard, and cassette player built into a single box. It used a microprocessor with a different architecture from the Intel 8080—the 6502 (sold by MOS Technologies). The PET's chief drawback was its calculator-style keyboard, and for that reason it was not as successful in the U.S. as the other comparable computers introduced that year. But it sold very well in Europe, and on the Continent it became a standard for many years.

The third machine introduced in 1977 was the Apple II (figure 8.5). The legend of its birth in a Silicon Valley garage, assisted by two idealistic young men, Steve Jobs and Steve Wozniak, is part of the folklore of Silicon Valley. According to the legend, Steve Wozniak chose the 6502 chip for the Apple simply because it cost less than an 8080. Before designing the computer he had tried out his ideas in discussions at the Homebrew Computer Club, which met regularly at a hall on the Stanford campus. The Apple II was a tour de force of circuit design. It used fewer chips than the comparable Altair machines, yet it outperformed most of them. It had excellent color graphics capabilities, better than most mainframes or minicomputers. That made it suitable for fast-action interactive games, one of the few things that all agreed personal computers were good for. It was attractively housed in a plastic case. It had a nonthreatening, nontechnical name. Even though users had to open the case to hook up a printer, it was less intimidating than the Altair line of computers. Jobs and Wozniak, and other members of the Homebrew Computer Club, did not invent the personal computer, as the legend often goes. But the Apple II came closest to Stewart Brand's prediction that computers would not only come to the people, they would be embraced by the people as a friendly, nonthreatening piece of technology that could enrich their personal lives. The engineering and design of the Apple II reflected those aims.

Wozniak wrote his own BASIC for the Apple, but the Apple II was later marketed with a better version, written by Microsoft for the 6502 and

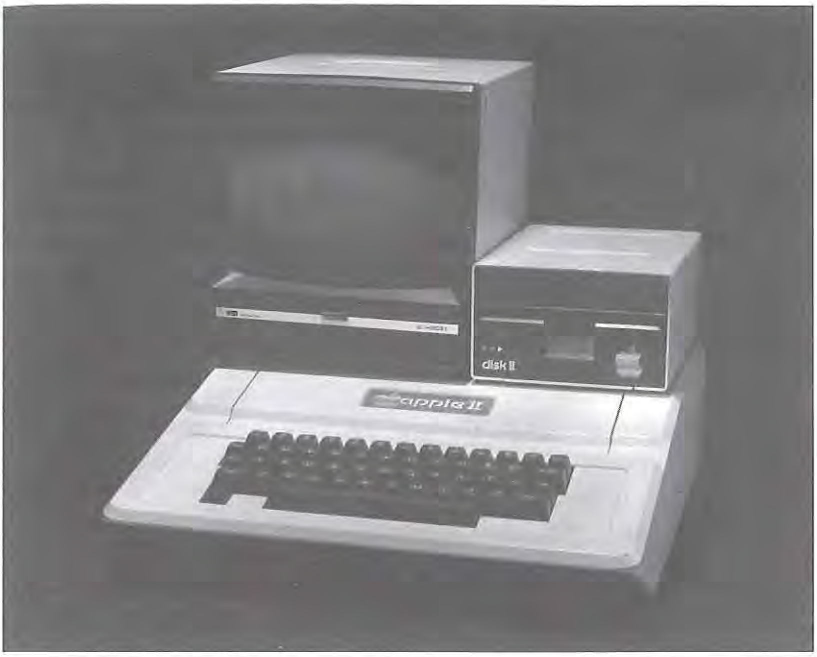

Figure 8.5
Personal computers: Apple II, ca. 1977, with a monitor and an Apple disk drive. (*Source:* Smithsonian Institution.)

supplied in a ROM. A payment of $10,500 from Apple to Microsoft in August 1977, for part of the license fee, is said to have rescued Microsoft from insolvency at a critical moment of its history.[57] Although it was more expensive than either the TRS-80 or the PET, the Apple II sold better. It did not take long for people to write imaginative software for it. Like the Altair, the Apple II had a bus architecture with slots for expansion—a feature Wozniak argued strenuously for, probably because he had seen its advantages on a Data General Nova.[58] The bus architecture allowed Apple and other companies to expand the Apple's capabilities and keep it viable throughout the volatile late 1970s and into the 1980s. Among the cards offered in 1980 was the SoftCard, from Microsoft, which allowed an Apple II to run CP/M. For Microsoft, a company later famous for software, this piece of hardware was ironically one of its best selling products at the time.

By the end of 1977 the personal computer had matured. Machines like the TRS-80 were true appliances that almost anyone could buy and get running. They were useful for playing games and for learning the rudiments of computing, but they were not good enough for serious applications. Systems based on the Altair bus were more sophisticated and more difficult to set up and get running, but when properly configured could compete with minicomputers for a variety of applications. The Apple II bridged those two worlds, with the flexibility of the one and the ease of use and friendliness of the other. At the base was a growing commercial software industry.

None of this was much of a threat to the computer establishment of IBM, Digital, Data General, or the BUNCH. Within a few years, though, the potent combination of cheap commodity hardware and commercial software would redefine the computer industry and the society that would come to depend on it. The trajectories of DEC, IBM, Wang, and Xerox did not intersect those of MITS, IMSAI, Apple, Radio Shack, or the other personal computer suppliers into the late 1970s. Innovations in personal computing did not seem as significant as those at places like Xerox or even IBM. But in time they would affect all of computing just as much. One of those innovations came from Apple.

APPLE II's Disk Drive and VisiCalc

By 1977 many personal computer companies, including MITS and IMSAI, were offering 8-inch floppy disk drives. These were much better than cassette tape but also expensive. The Apple II used cassette tape, but by the end of 1977 Steve Wozniak was designing a disk controller for it. Apple purchased the drives themselves (in a new 5 1/4-inch size) from Shugart Associates, but Wozniak felt that the controlling circuits then in use were too complex, requiring as many as fifty chips. He designed a circuit that used five chips. It was, and remains, a marvel of elegance and economy, one that professors have used as an example in engineering courses. He later recounted how he was driven by aesthetic considerations as much as engineering concerns to make it simple, fast, and elegant.[59]

Apple's 5 1/4-inch floppy drive could hold 113 Kbytes of data and sold for $495, which included operating system software and a controller that plugged into one of the Apple II's internal slots.[60] It was a good match for the needs of the personal computer—the drive allowed people to market and distribute useful commercial software, and not just the simple games and checkbook-balancing programs that were the limit of cassette tape capacity. Floppy disk storage, combined with operating system software that insulated software producers from the peculiarities of specific machines, brought software to the fore. Ensuing decades would continue to see advances in hardware. But no longer would computer generations, defined by specific machines and their technology, best describe the evolution of computing. With a few exceptions, new computers would cease to be pivotal—or even interesting—to the history of computing.

In October 1979 a program called VisiCalc was offered for the Apple II. Its creators were Daniel Bricklin and Robert Frankston, who had met while working on Project MAC at MIT. Bricklin had worked for Digital Equipment Corporation and in the late 1970s attended the Harvard Business School. There he came across the calculations that generations of B-school students had to master: performing arithmetic on rows and columns of data, typically of a company's performance for a set of months, quarters, or years. Such calculations were common throughout the financial world, and had been semi-automated for decades using IBM punched-card equipment. He recalled one of his professors posting, changing, and analyzing such tables on the blackboard, using figures that his assistant had calculated by hand the night before. Bricklin conceived of a program to automate these "spreadsheets" (a term already in limited use among accountants). Dan Flystra, a second-year student who had his own small software marketing company, agreed to help him market the program. Bricklin then went to Frankston, who agreed to help write it.

In January 1979 Bricklin and Frankston formed Software Arts, based in Frankston's attic in Arlington, Massachusetts (the Boston area has fewer garages than in Silicon Valley). That spring the program took shape, as Frankston and Bricklin rented time on the MIT Multics system. In June, VisiCalc was shown at the National Computer Conference. The name stood for visible calculator, although inspiration for it may have come from eating breakfast one morning at Vic's Egg on One coffee shop on Massachusetts Avenue. (Nathan Pritikin would not have approved, but such eateries are another common feature of the Boston scene not found in Silicon Valley.)[61]

Bricklin wanted to develop this program for DEC equipment, "and maybe sell it door-to-door on Route 128." Flystra had an Apple II and a TRS-80; he let Bricklin use the Apple, so VisiCalc was developed on an Apple. The price was around $200. Apple itself was not interested in marketing the program. But the product received good reviews. A financial analyst said it might be the "software tail that wags the hardware dog."[62] He was right: in many computer stores people would come in and ask for VisiCalc and then the computer (Apple II) they needed to run it. Sales passed the hundred thousand mark by mid-1981 (the year the IBM personal computer was announced, an event that led to Software Arts's demise).

An owner of an Apple II could now do two things that even those with access to mainframes could not do. The first was play games; admittedly not a serious application, but one that nevertheless had a healthy market. The second was use VisiCalc; which was as important as any application running on a mainframe. Word processing, previously available only to corporate customers who could afford systems from Wang or Lanier, soon followed.

IBM PC (1981)

Although after the Apple II and its floppy drive were available, one could say that hardware advances no longer drove the history of computing, there were a few exceptions, and among them was the IBM Personal Computer. Its announcement in August 1981 did matter, even though it represented an incremental advance over existing technology. Its processor, an Intel 8088, was descended from the 8080, handling data internally in 16-bit words (external communication was still 8 bits).[63] It used the ASCII code. Its 62-pin bus architecture was similar to the Altair's bus, and it came with five empty expansion slots. Microsoft BASIC was supplied in a read-only memory chip. It had a built-in cassette port, which, combined with BASIC, meant there was no need for a disk operating system. Most customers wanted disk storage, and they had a choice of three operating systems: CP/M-86, a Pascal-based system designed at the University of California at San Diego, and PC-DOS from Microsoft. CP/M-86 was not ready until 1982, and few customers bought the Pascal system, so PC-DOS prevailed. The floppy disk drives, keyboard, and video monitor were also variants of components used before. IBM incorporated the monitor driver into the PC's basic circuit board, so that users did not tie up a communication port. The monochrome monitor could display a full screen of 25 lines of 80 characters—an improvement over the Apple II and essential for serious office applications. A version with a color monitor was also available (figure 8.6).

With the PC, IBM also announced the availability of word processing, accounting, games software, and a version of VisiCalc. A spreadsheet introduced in October 1982, 1-2-3 from Lotus Development, took

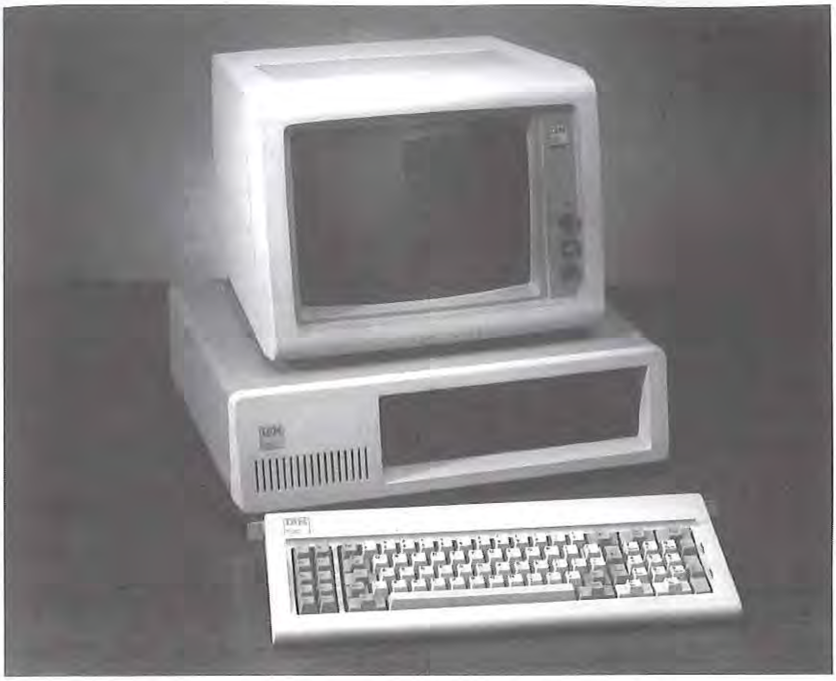

Figure 8.6
Personal computers: IBM PC, 1981. Note the two internal floppy disk drives. (*Source:* Smithsonian Institution.)

advantage of the PC's architecture and ran much faster than its competitor, VisiCalc. This combination of the IBM Personal Computer and Lotus 1-2-3 soon overtook Apple in sales and dispelled whatever doubts remained about these machines as serious rivals to mainframe and minicomputers. In December 1982 *Time* magazine named the computer "Machine of the Year" for 1983.[64]

MS-DOS

Microsoft was a small company when an IBM division in Boca Raton, Florida, embarked on this project, code named "Chess." Microsoft was best known for its version of BASIC. IBM had developed a version of BASIC for a product called the System/23 Datamaster, but the need to reconcile this version of BASIC with other IBM products caused delays. The Chess team saw what was happening in the personal computer field, and they recognized that any delays would be fatal. As a result they would go outside the IBM organization for nearly every part of this product, including the software.[65]

Representatives of IBM approached Bill Gates in the summer of 1980 to supply a version of BASIC that would run on the Intel 8088 that IBM had chosen.[66] IBM thought it would be able to use a version of CP/M for the operating system; CP/M was already established as the standard for 8080-based systems, and Digital Research was working on a 16-bit extension. But negotiations with Gary Kildall of Digital Research stalled. When IBM visited Digital Research to strike the deal, Kildall was not there, and his wife, wbo handled the company's administrative work, refused to sign IBM's nondisclosure agreement. (Given the charges that had been leveled against IBM over the years, she was not being unreasonable.[67]) In any event, Digital Research's 16-bit version of CP/M was not far enough along in development, although the company had been promising it for some time. (It was eventually offered for the IBM PC, after PC-DOS had become dominant.)

In the end, Microsoft offered IBM a 16-bit operating system of its own. IBM called it PC-DOS, and Microsoft was free to market it elsewhere as MS-DOS. PC-DOS was based on 86-DOS, an operating system that Tim Paterson of Seattle Computer Products had written for the 8086 chip. Microsoft initially paid about $15,000 for the rights to use Seattle Computer Products's work. (Microsoft later paid a larger sum of money for the complete rights.) Seattle Computer Products referred to it internally by the code name QDOS for "Quick and Dirty Operating System"; it ended up as MS-DOS, one of the longest-lived and most-influential pieces of software ever written.[68]

MS-DOS was in the spirit of CP/M. Contrary to folklore, it was not simply an extension of CP/M written for the advanced 8086 chip. Paterson was familiar with a dialect of CP/M used by the Cromemco personal computer, as well as operating systems offered by Northstar and a few other descendants of the Altair. A CP/M users manual was another influence, although Paterson did not have access to CP/M source code. Another influence was an advanced version of Microsoft BASIC that also supported disk storage, which it was probably led to the use of a file allocation table by MS-DOS to keep track of data on a disk. The 86-DOS did use the same internal function calls as CP/M; actually, it used 8086 addresses and conventions that Intel had published in documenting the chip, to make it easy to run programs written for the 8080 on the new microprocessor. It used the CP/M commands "Type," "Rename," and

"Erase." MS-DOS also retained CP/M's notion of the BIOS, which allowed it to run on computers from different manufacturers with relatively minor changes.[69]

It is worth mentioning the differences between CP/M and MS-DOS, since these help explain the latter's success. A few changes were relatively minor: the cryptic all-purpose PIP command was changed to more prosaic terms like "Copy"; this made MS-DOS more accessible to a new generation of computer users but severed the historical link with the Digital Equipment Corporation, whose software was the *real* ancestor of personal computer systems. CP/M's syntax specified the first argument as the destination and the second as the source; this was reversed to something that seems to be more natural to most people. (The CP/M syntax was also used by Intel's assembler code and by the assembler for the IBM System/360).[70] More fundamental improvements included MS-DOS's ability to address more memory—a consequence of the Intel chip it was written for. MS-DOS used a file allocation table; CP/M used a less-sophisticated method. CP/M's annoying need to reboot the system if the wrong disk was inserted into a drive was eliminated. Doing that in MS-DOS produced a message, "Abort, Retry, Fail?" This message would later be cited as an example of MS-DOS's unfriendly user interface, but those who said that probably never experienced CP/M's "Warm Boot" message, which was much worse and sometimes gave the feeling of being kicked by a real boot. Several features may have been inspired by UNIX, for example, version 2, which allowed users to store files on a disk in a hierarchical tree of directories and subdirectories.[71] Tim Paterson later stated that he had intended to incorporate multitasking into DOS, but "they [Microsoft] needed to get something really quick."[72]

System software, whether for mainframes or for personal computers, seems always to require "mythical man-months" to create, to come in over budget, and to be saddled with long passages of inefficient code. Tim Paterson's initial work on 86-DOS took about two months, and the code occupied about 6 K.[73] MS-DOS was, and is, a piece of skillful programming. It was the culmination of ideas about interactive computing that began with the TX-0 at MIT. It has its faults, some perhaps serious, but those who claim that MS-DOS's success was solely due to Bill Gates's cunning, or to Gary Kildall's flying his airplane when IBM's representatives came looking for him, are wrong.

The PC and IBM

The Personal Computer was IBM's second foray into this market, after the 5100—it even had the designation 5150 in some product literature. Neither IBM nor anyone else foresaw how successful it would be, or that others would copy its architecture to make it the standard for the next decade and beyond. In keeping with a long tradition in the computer industry, IBM grossly underestimated sales: it estimated a total of 250,000 units; "[a]s it turned out, there were some *months* when we built and sold nearly that many systems.[74] MS-DOS transformed Microsoft from a company that mainly sold BASIC to one that dominated the small systems industry in operating systems. IBM found itself with an enormously successful product made up of parts designed by others, using ASCII instead of EBCDIC, and with an operating system it did not have complete rights to. It was said that if IBM's Personal Computer division were a separate company, it would have been ranked #3 in the industry in 1984, after the rest of IBM and Digital Equipment Corporation. Within ten years there were over fifty million computers installed that were variants of the original PC architecture and ran advanced versions of MS-DOS.[75]

"The Better is the Enemy of the Good"

The evolution of technological artifacts is often compared to the evolution by natural selection of living things. There are many parallels, including the way selective forces of the marketplace affect the survival of a technology.[76] There are differences, too: living things inherit their characteristics from their parents—at most two—but an inventor can borrow things from any number of existing devices. Nor does nature have the privilege that Seymour Cray had, namely, to start with a clean sheet of paper when embarking on a new computer design.

The history of personal computing shows that these differences are perhaps less than imagined. The IBM PC's microprocessor descended from a chip designed for a terminal, although Datapoint never used it for that. Its operating system descended from a "quick and dirty" operating system that began as a temporary expedient. The PC had a limit of 640 K of directly addressable memory. That, too, was unplanned and had nothing to do with the inherent limits of the Intel microprocessor. 640 K was thought to be far more than adequate; within a few years that limit became a millstone around the necks of programmers and users alike. The IBM PC and its clones allowed commercial software to come to the fore, as long as it could run on that computer or machines that were 100 percent compatible with it. Those visionaries who had predicted and longed for this moment now had mixed feelings. This was what they wanted, but they had not anticipated the price to be paid, namely, being trapped in the architecture of the IBM PC and its operating system.

Macintosh (1984)

Among those who looked at the IBM PC and asked why not something better were a group of people at Apple. They scoffed at its conservative design, forgetting that IBM had made a deliberate decision to produce an evolutionary machine. They saw the limitations of MS-DOS, but not its value as a standard. (Of course, neitber did IBM at the time.) But what would personal computing be like if it incorporated some of the research done in the previous decade at Xerox's Palo Alto Research Center? The Xerox Star had been announced within months of the PC, but it failed to catch on. Some people at Apple thought they could be more successful.

For all the creative activity that went on at Xerox-PARC in the 1970s, it must be emphasized that the roots of personal computing—the microprocessor, the Altair, the bus architecture, the Apple II, BASIC, CP/M, VisiCalc, the IBM PC, the floppy disk, Lotus 1-2-3, and MS-DOS—owed *nothing* to Xerox-PARC research.

In 1979 that began to change. That fall Apple began work on a computer called the Macintosh. It was the brainchild of Jef Raskin, who before joining Apple had been a professor of computer science at UC San Diego. He had also been the head of a small computer center, where he taught students to program Data General Novas.[77] Raskin had also been a visiting scholar at Stanford's Artificial Intelligence Laboratory, and while there he became familiar with what was going on at Xerox-PARC. According to Raskin, he persuaded the Apple team then developing another text-based computer to incorporate the graphics features he had seen at PARC. Apple introduced that computer, the Lisa, in 1983. Like the Xerox Star, it was expensive (around $10,000), and sales were disappointing. Raskin's Macintosh would preserve the Lisa's best features but sell at a price that Apple II customers could afford.[78] As with so much in the history of computing, there is a dispute over who was responsible for the Macintosh.[79] Many histories describe a visit by

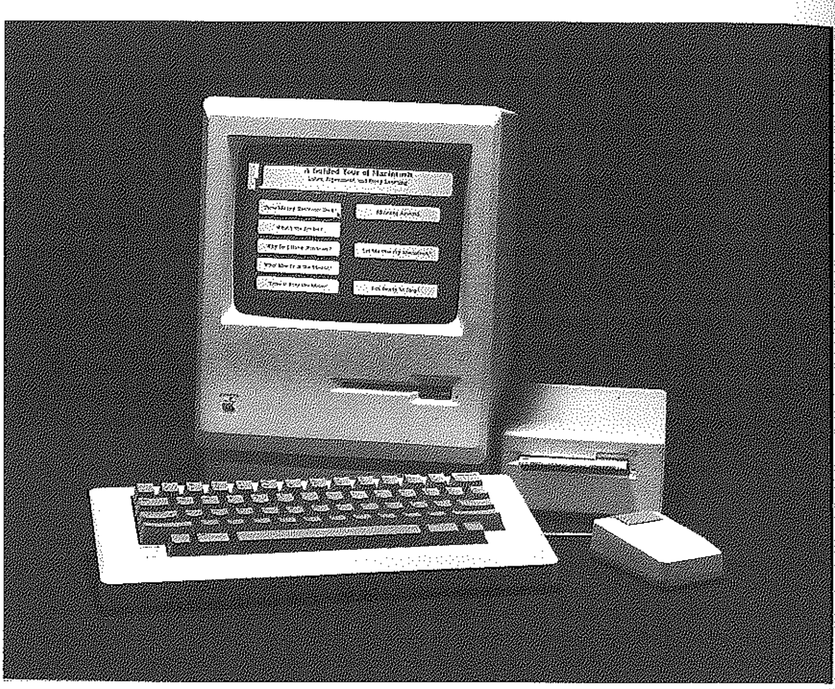

Figure 8.7
Personal computers: Apple Macintosh, 1984. Most Macintosh users soon found that the machine required a second, external disk drive. (*Source:* Smithsonian Institution.)

Apple cofounder Steve Jobs to PARC in 1979 as the pivotal moment in transferring PARC technology to a mass market. Work on the Macintosh was already underway at Apple by the time of that visit. The visit did result in Jobs' hiring several key people away from Xerox, however, and moving people is the best way to transfer technology. According to Raskin, the visit also resulted in Jobs' insisting that the Macintosh have features not present in the original design. Among those was the mouse (figure 8.7).[80]

In January 1984 Apple introduced the Macintosh in a legendary commercial during the Super Bowl, in which Apple promised that the Macintosh would prevent the year 1984 from being the technological dystopia forecast by Orwell's novel *1984*. The computer sold for $2,495—more than the $1,000 Raskin was aiming for, but cheaper than the Lisa. It was more expensive than an IBM PC, but no PC at that time, no matter what software or boards users added, could offer the graphical interface of the Macintosh.

The Macintosh used a Motorola 68000 microprocessor, whose architecture resembled that of the PDP-11. The computer came with a single disk drive, using the new 3 1/2-inch form, a high-resolution black-on-white monitor, a mouse, and 128K of memory. Most users found they soon had to upgrade to a 512K "Fat Mac"; they also found it necessary to purchase a second disk drive. A few programs were announced at the same time: a "paint" (drawing) program, based on work done at Xerox-PARC on a Data General Nova, and a word processor that came close to WYSIWYG.

A year later the Macintosh came with a rudimentary networking ability, called AppleTalk. This allowed the simple sharing of files and printers. Like so much about the system, it was simple, easy to use, and not challenged by the PC and its clones for years. But there was no hard disk option, so users could not effectively set up a Mac as a server to the others. A person using a Macintosh at home would not be connected to a network, and the Mac was unable to challenge the lead of IBM and its clones in an office environment, except in those offices where the graphics abilities were especially needed. Unlike the Apple II and the IBM PC, the Macintosh was "closed": users could not add boards and were discouraged from even opening up the case.[81] This was a bold—some argued foolish—departure from the prevailing wisdom, but it helped make the Macintosh cheaper, smaller, and faster than the Lisa or the Star. A version introduced in 1987 offered color and opened up the system, although Apple still tightly controlled the Macintosh's configuration.[82]

The Mac's elegant system software was its greatest accomplishment. It displayed a combination of aesthetic beauty and practical engineering that is extremely rare. One can point to specific details. When a file was opened or closed, its symbol expanded or contracted on the screen in little steps—somehow it just felt right. Ultimately this feeling is subjective, but it was one that few would disagree with. The Macintosh software was something rarely found among engineering artifacts. The system evolved as the Mac grew, and it was paid the highest compliment from Microsoft, who tried to copy it with its Windows program. One can hope that some future system will have that combination as well, but the odds are not in favor of it.

The Macintosh had more capability than the Alto, it ran faster than the Lisa, yet its software occupied a fraction of the memory of either of those predecessors. It was not just a copy of what Xerox had done at PARC. But there was a price for being so innovative: the Macintosh was difficult for programmers to develop applications software for, especially compared to MS-DOS. And though faster than the Lisa, its complex graphics meant that it could not he as fast as a DOS program, like Lotus 1-2-3, that used more primitive commands that were closer to machine code. Among sophisticated customers that created a split: one group favored the elegance and sophistication of the Mac, while others preferred the raw horsepower and access to individual bits that MS-DOS allowed. For those who were not members of the computer priesthood, the Macintosh was a godsend; whatever time was lost by its relative slowness was more than compensated for by the time the user did not have to spend reading an indecipherable users manual.

Microsoft had supplied some of the applications software for the Macintosh, but Apple developed and controlled its operating system in-house. Even before the Macintosh's announcement, other companies were trying to provide a similar interface for the IBM PC. In 1982 the creators of VisiCalc announced a product called VisiOn for the IBM PC that was similar to the Macintosh's interface but never lived up to its promise. IBM developed a program called Top View, and Digital Research developed GEM (Graphics Environment Manager) along the same lines. Microsoft came up with a product called Interface Manager, but early versions introduced in the mid-1980s sold poorly. Later versions of Interface Manager, renamed "Windows," would succeed dramatically. Version 3 of Windows, the breakthrough version, was not introduced until around 1990, so for the next seven years, IBM PCs and their clones would be known by the primitive MS-DOS interface inherited from the minicomputer world.

Like the IBM PC, the Macintosh's design created a barrier to expanding memory, only it was a more generous 4 megabytes instead of the PC's miserly 640 Khytes. A laser printer offered in 1985 completed the transfer of Xerox-PARC innovations and allowed the Macintosh to keep a strong foothold in at least some offices. The Macintosh's equivalent of VisiCalc was a program called PageMaker from Aldus, introduced in 1985. When comhined with the laser printer it allowed users to do sophisticated printing on an Apple, at a fraction of the cost of traditional methods.

The Clones

The personal computer revolution seems to have little to do with the age of mainframes that preceded it, but with the passage of time, we can find common themes. IBM's success with its System/360, and its need to give out a lot of technical information about it, led to the plug compatible industry, which in turn led to IBM's having to adjust its own product line. Something similar happened with the PC, only this time with a different outcome. Most of the IBM PCs, including the 8088 microprocessor, consisted of parts made by other manufacturers, who were free to sell those parts elsewhere. Microsoft, for instance, retained the right to sell its operating system to others. The core of what made a personal computer an "IBM PC" was the basic input-output system (BIOS), which was stored on a read-only memory chip. The idea went back to Gary Kildall's CP/M: let the BIOS be the only place where there could be code that tailored the operating system to the specifics of a particular machine. IBM owned the code in the personal computer's BIOS and prosecuted any company that used it without permission.

Around the time of the PC's announcement, three Texas Instruments employees were thinking of leaving their jobs and starting a company of their own, which they called Compaq. Legend has it that Rod Canion, Jim Harris, and Bill Murto sketched out an IBM-compatible PC on a napkin in a Houston restaurant. They conceived of the idea of reverse-engineering the IBM PC and producing a machine that would be 100 percent compatible. To get around IBM's ownership of the BIOS code, they hired people who had no knowledge of that code, put them in a "clean room," where they would not be corrupted by anyone sneaking the forbidden code to them, and had them come up with a BIOS of their own that replicated the functions of IBM's. This was expensive, but it was legal. The Compaq computer, delivered in 1983, was portable, although heavy. That was really a marketing ploy: At twenty-five pounds they "gave new meaning to the phrase pumping iron." What made it a success was its complete compatibility with the IBM PC at a competitive price. Compaq's sales propelled the company into the top 100 rankings of computer companies by 1985, one of the fastest trajectories of any start-up.[83]

Compaq's heroic efforts to break through IBM's control of its PC architecture did not have to be repeated too often. A small company named Phoenix Technologies also reverse-engineered the BIOS chip, and instead of building a computer around it, they simply offered a

BIOS chip for sale. Now building an IBM-compatible PC was easy. The trade press instituted a test for compatibility: would the machine run Lotus 1-2-3, which was written to take advantage of the PC's inner workings to gain maximum speed? Better still, would it run Flight Simulator, a program written by Bruce Artwick that exercised every nook and cranny of the IBM architecture?[84] If the answer was Yes and Yes, the machine was a true clone. The floodgates opened. Unlike its successful footwork during the times of System/360 and the plug compatibles, this time IBM lost control over its own architecture.

The introduction of IBM Compatibles and the Macintosh signaled the end of the pioneering phase of personal computing. Minicomputer and mainframe manufacturers could no longer ignore this phenomenon. In the late 1980s, companies like Novell would introduce more capable networking abilities for personal computers, which allowed networks of PCs to seriously challenge many large systems. After some hesitant

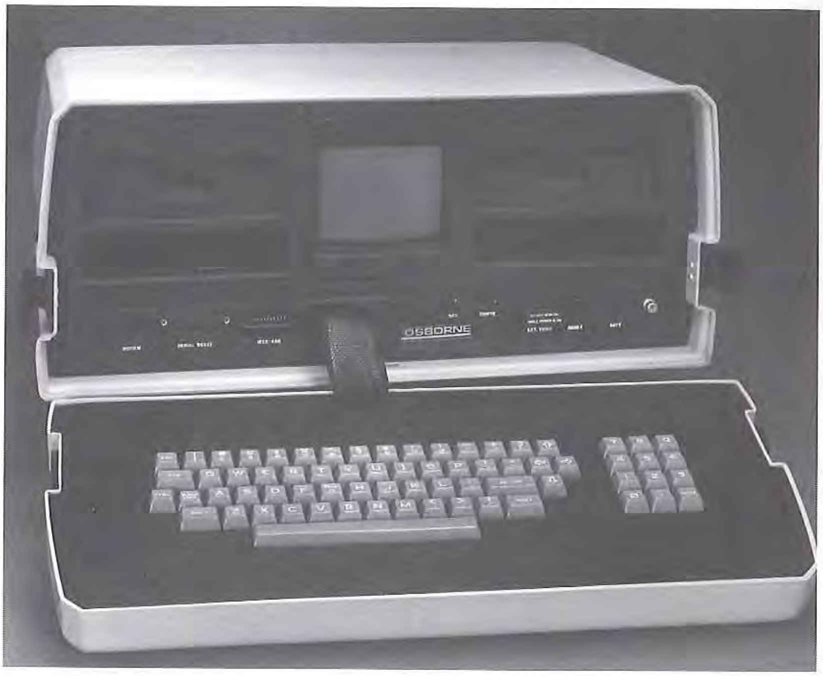

Figure 8.8
An early "transportable" computer. Osborne, ca. 1981. Just as revolutionary as its small size was the fact that the computer came with the CP/M operating system and applications software, all for less than $2,000.

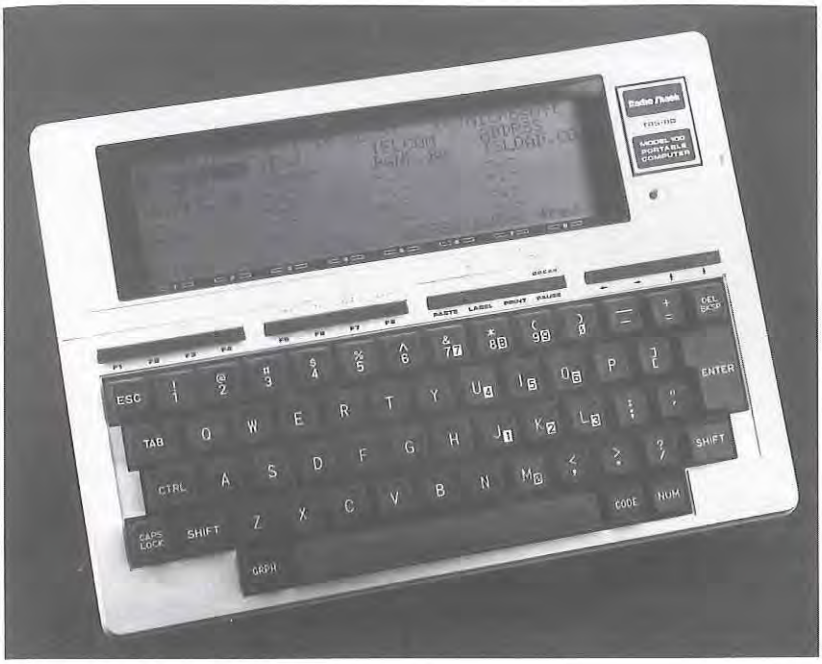

Figure 8.9
An early "laptop" computer. Tandy Radio Shack TRS-80, Model 100, ca. 1983. Like the Osborne, it used an 8-bit microprocessor. System software and the BASIC programming language were supplied by Microsoft and included with the machine. The machine shown here was much modified and extended and served as the author's home computer for many years. (*Source:* Smithsonian Institution.)

beginnings based on 8-bit designs, manufacturers developed portable computers that were compatible with those on the desktop (figs. 8.8, 8.9). Commercial software, driven relentlessly by the marketplace created by Microsoft, led to applications that likewise challenged the mini and mainframe world. By 1991 the IBM-compatible computers, based on advanced versions of the Intel 8086 chip and running Windows 3.1, brought the Macintosh's features to the business and commercial world. For reasons having to do more with IBM's poor management than anything else, companies like Compaq and Dell would earn more profits selling IBM-compatible computers than IBM would. IBM remained a major vendor, but the biggest winner was Microsoft, whose operating system was sold with both IBM computers and their clones.

The personal computer revolutionized the office environment, but it had not become a revolutionary machine in the political or cultural sense, the sense that Stewart Brand and others had predicted and hoped for. Computers came "to the people," but for a price: corporate control.