8 Computing gets personal

> I think it's fair to say that personal computers have become the most empowering tool we've ever created. They're tools of communication, they're tools of creativity, and they can be shaped by their user.
>
> Bill Gates[1]

The beginnings of interactive computing

In the early days of computing, computers were expensive and scarce. They were built for solving serious computational problems – and certainly not for frivolous activities like playing games! The microprocessor and Moore's law have changed this perspective – computing hardware is now incredibly cheap and it is the software production by humans and management of computers that is expensive. Some of the ideas of interactive and personal computing can be traced back to an MIT professor called J. C. R. Licklider. Lick – as he was universally known – was a psychologist and one of the first researchers to take an interest in the problem of human-computer interactions. During the Cold War in the 1950s, he had worked at MIT's Lincoln Labs on the Semi-Automated Ground Environment (SAGE) system designed to give early warning of an airborne attack on the United States. This system used computers to continuously keep track of aircraft using radar data. It was this experience of interactive computing that convinced Lick of the need to use computers to analyze data as the data arrived – for "real time" computing.

Another type of interactive computing was being developed at around the same time. Engineers at MIT's Lincoln Labs had developed the TX-0 in 1956 – one of the first transistorized computers. Wesley Clark and Ken Olsen had specifically designed and built the TX-0 to be interactive and exciting, the exact opposite of sedate batch processing on a big mainframe computer. Olsen recalled:

> Then we had a light pen, which was what we used in the air-defense system and which was the equivalent of the mouse or joystick we use today. With that you could draw, play games, be creative – it was very close to being the modern personal computer.[2]

This level of interactivity, and for what then seemed to be frivolous uses of valuable computing time, was a far cry from the regimented bureaucracy of batch processing. To popularize their ideas, Olsen and Clark decided to send

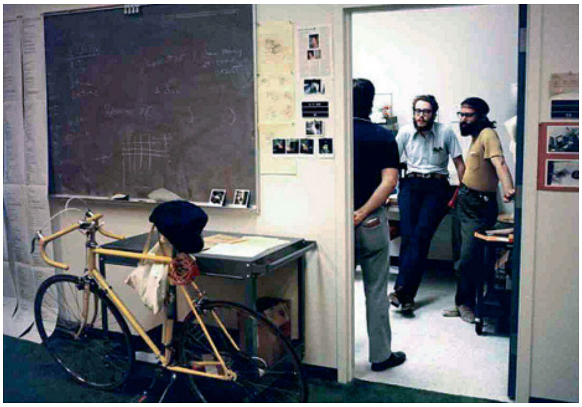


Fig. 8.1. Eager to be known as more than a supplier of office copiers, Xerox created PARC in 1970. PARC's mission was to create the "Office of the Future." George Pake and Bob Taylor assembled a team of world-class scientists and engineers – to create the "architecture of information" – and PARC became a hothouse of innovation that flourished for decades. The atmosphere at Xerox PARC reflected the laid-back, West Coast, hippie-influenced culture of the 1970s. It was worlds apart from the culture of Xerox's corporate headquarters in Connecticut. In an unrivaled burst of creativity, the PARC researchers developed most of the personal computing environment that is still with us today – overlapping windows, GUIs, Ethernet, digital video, word processing, and laser printers. Although PARC's inventions never led to a successful personal computer business for Xerox and many ideas never became successful commercial products, the laser printer alone generated billions of dollars in sales for Xerox, much more than their total investment in PARC.

the TX-0 from their off-campus Lincoln Laboratory site over to the main MIT campus. Clark later wrote:

> The only surviving computing system paradigm seen by MIT students and faculty was that of a very large International Business Machine in a tightly sealed Computation Center: the computer not as a *tool*, but as a *demigod*. Although we were not happy about giving up the TX-0, it was clear that making this small part of Lincoln's advanced technology available to a larger MIT community would be an important corrective step.[3]

Yet a third type of interactive computing was also being experimented with at MIT. As we have seen, John McCarthy had become so frustrated with this remote, batch processing model of computing that he had come up with the idea of time sharing. Sharing the computing cycles of a single large computer among several users, each connected to the computer with their own terminal, introduced a different type of interactivity – one in which the user had the illusion of being the sole user of the computer.

It was out of this hotbed of experimentation with interactive computing at MIT that Lick was recruited in 1962 to lead a new computer research program at the U.S. Department of Defense's Advanced Research Projects Agency (ARPA). When he arrived at the Pentagon, Lick set about creating a major research program in interactive computing – and in so doing laid the foundations for much of the university computer science research in the United States. As we will see in Chapter 10, Lick also had the idea of connecting remote computers together to create what later became the ARPANET – although it was left to Bob Taylor, one of Lick's successors at ARPA, to get the funding to implement Lick's vision.

How did we get from these early explorations of interactive computing to the personal computing we see around us today? In his book *Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age*, Michael Hiltzik highlights the contribution of the researchers at Xerox Corporation's Palo Alto Research Center (PARC) (Fig. 8.1):

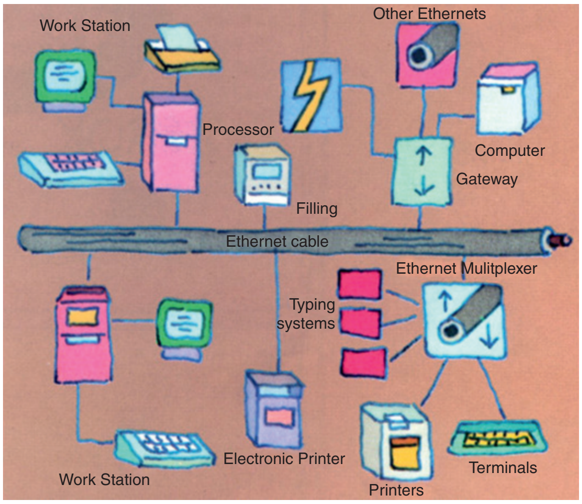


Fig. 8.2. A concept sketch for Metcalf and Bogg's Ethernet. Their original Ethernet report observed: "Just as computer networks have grown across continents and oceans to interconnect major computing facilities … they are now growing down corridors and between buildings to interconnect minicomputers in offices and laboratories."[F1]

> Every time you click a mouse on an icon or open overlapping windows on your computer screen today, you are using technology invented at PARC. Compose a document by word processor, and your words reach the display via software invented at PARC. Make the print larger or smaller, replace ordinary typewriter letters with a Braggadocio or Gothic typeface – that's technology invented at PARC, as is the means by which a keystroke speeds the finished document by cable or infrared link to a laser printer. The laser printer, too, was invented at PARC.[4]

Why then was Xerox not at the heart of the personal computer revolution? The answer is complicated, but ultimately Xerox failed to fully exploit the amazing inventions of its PARC researchers and missed a huge opportunity to create a new computing paradigm. Nevertheless, just one of PARC's inventions, the laser printer, generated billions of dollars in revenue for the company, many times more than its total investment in PARC (Fig. 8.2). But there could have been so much more (Fig. 8.3). This wonderful burst of creativity at Xerox PARC took place in the early 1970s. The personal computer revolution arrived by a different route and was triggered by the arrival of cheap and powerful microprocessors, an enthusiastic community of computer hobbyists, and four remarkable young entrepreneurs without a university degree between them!

The Altair and Microsoft

In January 1975, an editorial in the magazine *Popular Electronics* proudly announced the arrival of the world's first "home computer" (Fig. 8.4):

> For many years, we've been reading and hearing about how computers will one day be a household item. Therefore, we're especially proud to present in this issue the first commercial type of minicomputer project ever published

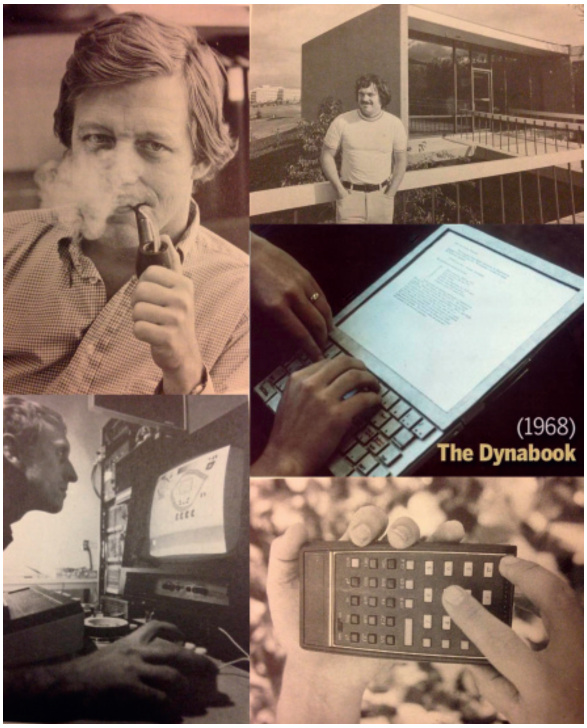


Fig. 8.3. Taylor and PARC scenes. Clockwise from top left: Bob Taylor, Alan Kay, the Dynabook, the pocket calculator, and *Rolling Stone* reporter Stewart Brand drawing with a computer.

Fig. 8.4. The launch of the Altair by *Popular Electronics* in January 1975. This cover featuring the MITS Altair 8800 excited thousands of hobbyists eager to build their own computers. Roberts had not finished the design of the computer when editor Les Solomon solicited the article. The magazine chose the name Altair after the star. The prototype machine sent by Ed Roberts got lost in the mail and never arrived so the magazine photographed an empty box for the cover photo.

> that's priced within reach of many households – the Altair 8800, with an under $400 complete kit cost, including cabinet.[5]

On the cover was a picture of the Altair 8800 computer, complete with flashing lights. In actual fact, it was just a picture of an empty case: the first prototype had gone missing in a shipment between Albuquerque and New York, and there had not been time to assemble another machine and get it to New York before the magazine's deadline. The Altair was the brainchild of Ed Roberts, a U.S. Air Force electronics engineer, and his small electronics hobby-kit company called Micro Instrumentation and Telemetry Systems (MITS), based in Albuquerque, New Mexico (Fig. 8.5). His company had been one of the first to put a calculator kit on the market, but by 1974 fully assembled calculators were selling for less than the kit. To save the company, Roberts devised a totally new product – a computer kit based around the latest 8080 microprocessor from Intel. This chip was more powerful than its predecessor, the 8008, and, as historian Paul E. Ceruzzi says, "[it] required only six instead of twenty supporting chips to make a functional system."[6] In his design Roberts introduced an "open bus architecture" to allow for the addition of extra circuit boards. A *bus* is just a set of connections linking all the major components of the machine, including the central processor, memory, and input/output (I/O) devices, in a standard way. The *bus architecture* makes it possible to customize the computer with circuit boards offering a specific functionality. If a user wants a better sound system, for example, he or she is able simply to unplug the old sound card from the bus and plug in a new one.

MITS planned to produce and sell plug-in cards for *peripheral devices*, auxiliary devices that would connect to the computer, such as memory boards, paper-tape readers, terminals, and printers. Making the bus design open was

Fig. 8.5. The MITS Altair 8800. Most do-it-yourself hobbyists did not want to have to buy all the integrated circuits and other components needed to build the computer. The appeal of the Altair 8800 was that it was the first build-your-own-computer kit.

important because it would allow hobbyists and other electronics companies to make cards for the Altair. In spite of the fact that no peripheral cards were available for many months after launch, and there was no prepackaged software available for the machine, customers deluged MITS with orders for the Altair. But the machine was far from being a household item: to get the Altair to actually do anything, a user had to painstakingly enter a program by hand, bit by bit, using toggle switches on the front panel. It was clear from the time of its launch that what the Altair urgently needed was the capability to run a high-level programming language.

The BASIC programming language was developed in the 1960s by John Kemeny and Thomas Kurtz at Dartmouth College. BASIC is an acronym for Beginner's All-purpose Symbolic Instruction Code, and the language was used at Dartmouth for teaching undergraduates. A team of engineers at Digital Equipment Corporation (DEC) had taken an important step in the evolution of BASIC in 1971. They used BASIC to implement a new operating system for the PDP-11 minicomputer, and they extended and modified the language in a number of important ways. The most important was the introduction of "PEEK" and "POKE" commands, which gave programmers the ability to execute low-level system calls, and to interact with the computer's memory at the byte level directly from a BASIC program. The engineers also made some compromises to the original rules of the language, changes disapproved of by Kemeny and Kurtz, that allowed DEC BASIC to be used on machines with very limited memory. Despite its ease of use, academic computer science departments often discouraged BASIC as a teaching language because it was believed to encourage bad programming habits. Edsger Djikstra, who received the Turing Award for his contributions to computing, went so far as to say that programming in BASIC causes brain damage (see B.4.11). For the personal computing revolution, BASIC, with the DEC extensions allowing programmers to pass easily from BASIC to machine code, was the obvious first choice. Ed Roberts said that he had settled on BASIC for the Altair because you "could teach any idiot how to use [it] in no time at all."[7]

Paul Allen and his friend from high school, Bill Gates, had been entranced by computers from their days at Lakeside School, a private school in Seattle (B.8.1). In their spare time, they had worked as testers for the C-Cubed computer

B.8.1. The photograph shows Paul Allen at a teletype and Bill Gates (standing) when they were at Lakeside School in Seattle in 1968. Allen and Gates had signed up for the school's independent study option on programming and learned to program in BASIC. The two became captivated by computing and spent many hours of their spare time working for a local computer company. As a result of this experience, they became proficient in operating system software and assembly language for the PDP-10. The appearance of the Altair do-it-yourself computer kit on the cover of the magazine *Popular Electronics* in January 1975 excited Allen and Gates. They contacted Ed Roberts, the designer of the Altair, and offered to produce a BASIC interpreter for the machine. Remarkably, they had no access to an Altair machine when they wrote their interpreter. Instead, they debugged their BASIC interpreter using a simulator of the Intel 8080 microprocessor that Allen wrote for the PDP-10 that Gates had access to at Harvard. Aided by fellow Harvard student Monte Davidoff, the three of them finished their BASIC interpreter in just eight weeks. When Harvard reviewed the usage statistics of their PDP-10 machine during January, they found that William Henry Gates III had used a surprisingly large amount of computer time!

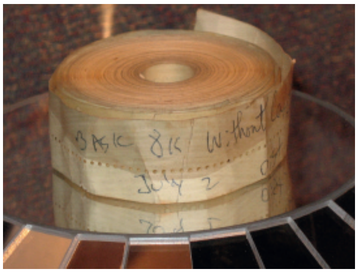


Fig. 8.6. The interpreter source tape for Altair BASIC. Paul Allen finished the software while flying to Albuquerque to demonstrate the interpreter to Ed Roberts and his engineers at MITS. Microsoft later created interpreters for many other languages and processors, although BASIC remained its most valuable product into the early 1980s. The text on the tape reads "BASIC 8K without cassette July 2 1975."

company in exchange for free use of the company's new PDP-10 minicomputer. There they learned new programming skills from the company's expert programmers. One of them, Steve Russell, had worked with John McCarthy at MIT and had developed Spacewar!, one of the first interactive computer games. In those early days, computer companies placed most value on their hardware: software came free as an inducement for customers to buy the machine. As a result, C-Cubed had access to the source code of the TOPS-10 operating system, developed by DEC for the PDP-10 mainframe computer, and C-Cubed was working to debug and enhance the system. Russell noticed Allen's interest in learning more about programming and introduced him to the PDP-10 assembly language. As a project, he suggested that Allen try to improve and enhance the BASIC compiler for the PDP-10. When C-Cubed closed down, Allen and Gates continued their projects by relocating unofficially to the computer science lab at the University of Washington. During the next few years, they also worked as programmers on various commercial contracts, writing code for PDP-10 machines. In the summer of 1972, they formed a partnership called Traf-O-Data to develop both the hardware and software to automate the measurement of traffic flows using Intel's newest microprocessor, the 8008, to do the data analysis. They persuaded Paul Gilbert, an engineering student at the University of Washington, to design and build the hardware. To write the software, because the hardware did not yet exist, they decided to simulate the 8008's instruction set on a PDP-10 minicomputer. Although Traf-O-Data was not a commercial success, Allen and Gates built an unrivaled set of development tools for the 8008 microprocessor. These tools included an assembler, to translate from assembly language into machine code; a simulator, to model and study real-life situations on the computer; and a debugger that allowed the programmer to stop the program in mid-execution.

By December 1974, Bill Gates had gone to Harvard, and Paul Allen had also moved to the Boston area working as a programmer for Honeywell. When Allen came across the January issue of *Popular Electronics*, the two friends realized that their experience had uniquely prepared them for the challenge of writing BASIC for the Altair. Allen describes Gates calling Ed Roberts (B.8.2) in Albuquerque, pretending to be Paul Allen:

> "This is Paul Allen in Boston," Bill said. "We've got a BASIC for the Altair that's just about finished, and we'd like to come out and show it to you." I admired Bill's bravado but worried that he'd gone too far, since we'd yet to write the first line of code.[8]

Roberts had received many calls from people making similar claims. He told Gates that he would give a contract to the first person to demonstrate a BASIC that actually worked on the Altair.

With this as encouragement, Allen and Gates bought an 8080 instruction manual and set about extending their Traf-O-Data development tools for the new microprocessor. Gates led the design of the BASIC *interpreter* (Fig. 8.6). A compiler converts the entire source code of a program into an assembly language program in one operation: an interpreter translates and executes small pieces of source code at a time and therefore takes up much less memory. To

B.8.2. Ed Roberts (1941–2010) founded the Micro Instrumentation and Telemetry Systems (MITS) company in 1970, and initially produced electronics kits for model rockets, and later, for calculators. When calculators became too cheap for the MITS kits to be profitable, Roberts designed a $397 "personal computer," do-it-yourself kit called the Altair 8800. After the January 1975 issue of *Popular Electronics*, orders began to pour in and the Altair became the catalyst for the personal computer revolution.

write the code for the decimal arithmetic operations required for BASIC, they recruited a fellow Harvard student named Monte Davidoff. That January and February, the three worked until late every night and through all the weekends. Their BASIC interpreter was finished in just eight weeks, and Allen flew to Albuquerque for its first encounter with the real Altair hardware. To the amazement of Roberts and his engineers, the 8080 BASIC interpreter developed by Gates and Allen ran the first time. The two friends signed a licensing agreement with MITS in July 1975 and needed a name for their partnership. They decided on Micro-Soft, for *Micro*processors and *Soft*ware, although they were not consistent about having the hyphen. In November 1976, the name of their company was registered with the state of New Mexico as Microsoft Corporation (B.8.3).

It was the unique technical experience of Allen and Gates, together with their PDP-10 simulator and development tools, that enabled them to beat seasoned professional software engineers and university computer scientists in developing the first usable software for the Altair. Their BASIC interpreter packed many features and impressive performance into a very small amount of memory. Paul Ceruzzi summarized their achievements:

> The BASIC they wrote for the Altair, with its skillful combination of features taken from Dartmouth and from Digital Equipment Corporation, was the key to Gates's and Allen's success in establishing a personal computer software industry.[9]

By 1979, Microsoft's BASIC interpreter became the first microprocessor software product to surpass a million dollars in sales (Fig. 8.7).

The Homebrew Computer Club and Apple

The arrival of the Altair stimulated the electronic hobbyist community to make microprocessor-based personal computers a reality. Computer clubs sprang up all over the United States including, most famously, the Homebrew Computer Club in Silicon Valley. In the early years of the personal computer,

B.8.3. This photograph of thirteen of the original fifteen Microsoft staff was taken in Albuquerque on 7 December 1978. Top row, left to right: Steve Wood, programmer; Bob Wallace, production manager-designer; Jim Lane, project manager. Middle row, left to right: Bob O'Rear, mathematician; Bob Greenberg, programmer; Marc McDonald, programmer and Microsoft's first employee; Gordon Letwin, programmer. Bottom row, left to right: Bill Gates, cofounder; Andrea Lewis, technical writer; Marla Wood, bookkeeper, married to Steve Wood; and Paul Allen, cofounder. Allen left Microsoft in 1983 and is now owner of the Seattle Seahawks, winners of the 2014 NFL Super Bowl. Two employees were not in the photograph. Ric Weiland was house hunting in preparation for Microsoft's move to Seattle, and Miriam Lubow was unable to make it into town for the photograph because of a rare snowstorm in Albuquerque that day.

Fig. 8.7. An aerial view of the present Microsoft campus in Redmond, near Seattle in Washington State.

from 1975 to 1978, hobbyists played a crucial role in its development, while the chip manufacturers and traditional computer companies focused on the business computer market. Chip suppliers were developing a market for microprocessors designed to handle control functions within larger systems – the *embedded systems* market. IBM, DEC, and other computer companies were focused on mainframes or mini-computers and had not embraced the idea of a truly personal computer. Only enthusiastic hobbyists were willing to put up with the difficulties of programming such primitive microprocessor systems like the Altair at a time when there were no peripheral devices available to make the system easier to use. Fortunately, the open bus architecture of the Altair meant that electronic hobbyists as well as other companies besides MITS were soon able to create these components and have a stake in this nascent industry.

Although IBM had started "unbundling" its hardware and software – that is, selling its hardware and software separately – as early as 1968, the original tradition of hardware manufacturers was for them to give the software away for free as an added feature of their machines. This practice led to a schism in the computing community that to some extent persists to this day. Allen and Gates were surprised and disappointed when they found that their royalty check for Altair BASIC in 1975 was only $16,005. Less than one in ten Altair owners was actually purchasing their BASIC software, instead relying on a tradition of widespread copying. This led to the famous "Open Letter to Hobbyists" from Bill Gates, published in the newsletter of the Homebrew Computer Club, in which Gates argued that unauthorized copying discouraged the development of high-quality software. The article generated a heated debate in the hobbyist community (Fig. 8.8).

The arrival of the Altair inspired the founding of the Homebrew Computer Club. The first meeting took place in March 1975 in a garage in Menlo Park, California, and subsequent monthly meetings were held in the auditorium of the Stanford Linear Accelerator Center. Among the thirty or so attendees at the first meeting – the numbers later grew to several hundred – was Stephen Wozniak, or Woz as he was known to his friends. Although he had dropped out of formal university education, Woz was an exceptionally talented computer engineer who worked in the calculator division of the Hewlett-Packard

Fig. 8.8. A meeting of the Homebrew Computer Club. The club met in the auditorium of the Stanford Linear Accelerator Center and hobbyists were encouraged to display their latest creations in the entry lobby. Anyone who attended even once was considered a "member" and could sign up for the newsletter. Founding member Fred Moore published the first issue of *Homebrew Computer Club Newsletter* on 15 March 1975. Moore expressed the shared excitement of the group: "I expect home computers will be used in unconventional ways – most of which no one has thought of yet."[F2]

Company in Palo Alto. Inspired by the Altair, Woz started building his own computer based on the 6502 microprocessor produced by MOS Technology, Inc. It was the cheapest fully functional microprocessor at the time, substantially undercutting the price of Intel's 8080. In six months, Woz had produced a circuit board for the 6502, with 4 kilobytes of memory and circuitry that allowed it to be directly connected to a monitor and keyboard. This was a great improvement in usability compared to toggling the switches on the Altair. He unsuccessfully tried to interest Hewlett-Packard, his employer, in commercializing it, but received an enthusiastic reception at the Homebrew Computer Club.

Fig. 8.9. Steve Wozniak demonstrated the prototype Apple I at the Homebrew Computer Club in 1976. For $666.66, buyers received a blank printed circuit board, a parts kit, and sixteen-page assembly manual. The power supply, keyboard, storage system, and display were not included.

In 1971, a friend had introduced Woz to teenager Steve Jobs, a fellow computer enthusiast (B.8.4). Together, Woz and Jobs designed and sold "blue boxes," unauthorized devices that enabled purchasers to mimic the control signals of the Bell Telephone Company's lines and make calls for free. After high school, Jobs went to Reed College in Portland, Oregon, but dropped out of full-time education and returned home to Los Altos, California. He went to work for Atari Inc., one of the first video game companies, until he had saved enough money to visit India to pursue his interest in Asian philosophy. When Jobs returned from India in 1974 he immediately saw the potential in Woz's personal computer board. Together with Ronald Wayne, Jobs and Wozniak founded the Apple Computer Company on 1 April 1976 to market the board that Woz had designed as a personal computer kit, later called the Apple I (Fig. 8.9). Wayne later sold his shares back to Jobs and Wozniak. Jobs persuaded the newly established Byte Shop store to order one hundred boards at $500 each. To get the funds to buy the chips and have the circuit boards manufactured, Jobs had to sell his Volkswagen van and Wozniak his programmable Hewlett-Packard calculator! They assembled the boards in the garage of Jobs's parents' home in Los Altos, and eventually managed to sell about two hundred computer kits and make a small profit. Jobs realized that the microprocessor-based computer could appeal to a much broader market than just computer enthusiasts if it was presented as a self-contained machine in a plastic case,

B.8.4. Steve Jobs (right) and Steve Wozniak met in a friend's garage in the late 1960s. The two of them bonded over their shared interest in electronics and practical jokes. Their first project together was to design, produce, and sell "blue boxes" that enabled the user to make long-distance telephone calls for free.

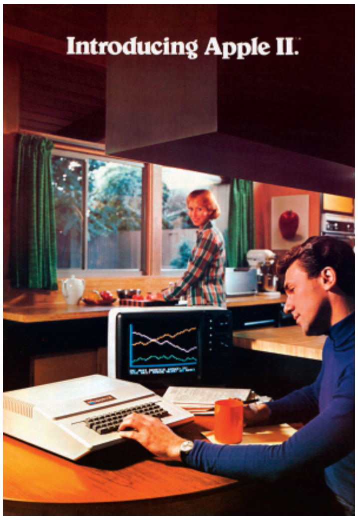


Fig. 8.10. When the Apple II was released in 1977, it was promoted as "an extraordinary computer for ordinary people."[F3] The self-contained system, user-friendly design, and graphical display made Apple a leader in the first decade of personal computing. Unlike the earlier Apple I, for which users had to supply essential parts such as a case and power supply, the Apple II was intended to be a fully realized consumer product. Apple's marketing emphasized its simplicity as an everyday tool for home, work, or school.

with a standard power supply, a keyboard and screen, and a cassette tape for long-term storage of data and programs. In addition, the computer would need a high-level programming interface and, potentially, a range of application software, including video games.

With this specification from Jobs, Woz set about creating the Apple II while Jobs set about getting the plastic cases made and raising start-up money (Fig. 8.10). Woz's design for the Apple II is recognized as a masterpiece of circuit design. It used fewer chips than the Altair, had good color graphics, and was great for the interactive games that Woz loved to play. Taking a page from the Altair playbook, Woz argued strenuously for the use of an open bus architecture with slots for expansion so that other companies could expand the machine's capabilities in interesting ways. He also wrote a version of BASIC for the machine. Meanwhile, Jobs had been introduced to Mike Markkula, only thirty-four at the time but already able to retire from his job as Intel's marketing manager with a considerable fortune generated by his Intel stock options. Markkula recognized the potential of the two young entrepreneurs and bought a third of the company and helped them write a business plan and raise venture capital. The Apple II was a great success and the advertising campaign claimed:

> The home computer that's ready to work, play and grow with you.… You'll be able to organize, index and store data on household finances, income taxes, recipes, your biorhythms, balance your checking account, even control your home environment.[10]

In reality, of course, there was no software to monitor your biorhythms, balance your checkbook, or perform any of these household applications at the time; most of the software available was still only for playing games.

For application software to really take off the personal computer needed a better and more convenient storage medium. Cassette tapes were slow and awkward, and could not provide random access; a user had to scroll through the tape from the beginning to reach any given point. These inconveniences disappeared with the invention of the *floppy disk* by David Noble of IBM in 1971. Floppy disks were flexible plastic disks coated with magnetic material that could be used to store information. IBM introduced the initial eight-inch floppies for loading the microcode for its mainframe computers. Alan Shugart, a former IBM manager whose team had helped develop the floppy disk, realized that this technology would be the ideal memory device for personal computers and set up a company to manufacture 5¼-inch floppy disks and disk drives. Although Apple purchased the drives from Shugart, Woz thought that the controlling circuits were too complex, requiring as many as fifty chips in total for their implementation. In another engineering *tour de force*, Woz redesigned the disk drive controller using only five chips and was able to deliver a floppy disk drive controller for the Apple II that was both simple and fast.

In 1979 the first "killer" business application for the personal computer emerged –an application that the Xerox PARC team had missed. This was the spreadsheet, a table used to present financial and other information. The first spreadsheet program was VisiCalc (Fig. 8.11), short for *Vis*ible *Calc*ulator. It was the brainchild of Daniel Bricklin (B.8.5), a twenty-six-year-old Harvard MBA

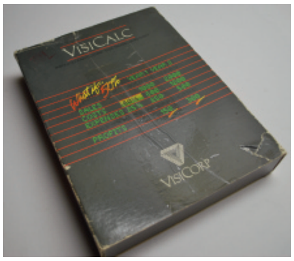


Fig. 8.11. The 1979 program, VisiCalc, was the first “killer” application for business. It was a spreadsheet program produced by Daniel Bricklin and Bob Frankston for their Software Arts company. Many customers bought an Apple computer specifically to run VisiCalc. Although VisiCalc was the first spreadsheet for personal computers, it was soon followed by other spreadsheet programs from Lotus, Microsoft, Borland, and others and eventually lost its supremacy in the market. Bricklin had not been able to patent the spreadsheet idea in VisiCalc because software patents were not generally issued until after a groundbreaking U.S. Supreme Court ruling in 1981.

student who had seen his fellow MBA students struggle to perform tedious and error-prone arithmetic operations on rows and columns of financial data. He conceived of VisiCalc as a program that would automate these spreadsheet calculations, and with Robert Frankston, who had worked with him on Project MAC at MIT, he set up a company to develop and market his new VisiCalc application. Although Apple was not interested in marketing the program directly, VisiCalc rapidly became a word-of-mouth success. As Robert Slater says in his book *Portraits in Silicon*:

> Suddenly it became obvious to businessmen that they had to have a personal computer: VisiCalc made it feasible to use one. No prior technical training was needed to use the spreadsheet program. Once, both hardware and software were for hobbyists, the personal computer a mysterious toy, used if anything for playing games. But after VisiCalc the computer was recognized as a crucial tool.[11]

Apple was incorporated in January 1977. When it went public in December 1980, it was one of the most successful stock offerings in Wall Street history and Jobs and Wozniak became multimillionaires overnight.

Project Chess and the IBM PC

By 1980, IBM had observed the rise of the Apple II and other microprocessor-based computers. A small group of advocates within the company realized that IBM could only become a dominant player in this emerging personal computer market if it could produce a machine very quickly. According to the

B.8.5. Bob Frankston (left) and Dan Bricklin, pioneers of the VisiCalc spreadsheet. Bricklin graduated from MIT in 1973 with a degree in electrical engineering and computer science. After some years in industry, he signed up for an MBA from Harvard Business School. It was while Bricklin was sitting in room 108 in Aldrich Hall at Harvard in 1978 that he dreamed of an easier way to calculate financial projections for multiple different business scenarios: “Imagine if my calculator had a ball in its back, like a mouse. . . .”[B1] He wrote a first prototype for the Apple II, which introduced rows and columns and some arithmetic operations. With fellow MIT graduate Bob Frankston, Bricklin founded Software Arts, Inc. in 1979 and began selling VisiCalc for $100 a copy. There is a plaque on the wall of Aldrich 108 commemorating Bricklin’s achievement: “In this room in 1978, Dan Bricklin, MBA ’79 conceived of the first spreadsheet program. VisiCalc, original ‘killer App’ of the information age, forever changed how people use computers in business.”[B2]

IBM archives, Bill Lowe and Don Estridge (B.8.6) of the IBM lab in Boca Raton, Florida, suggested the timescale required:

> One analyst was quoted as saying that "IBM bringing out a personal computer would be like teaching an elephant to tap dance." During a meeting with top executives in New York, Lowe claimed his group could develop a small new computer within a year. The response: "You're on. Come back in two weeks with a proposal."[12]

It was a controversial decision for IBM to enter the personal computer business. One insider was even reported as saying:

> Why on earth would you care about the personal computer? It has nothing at all to do with office automation. It isn't a product for big companies that use "real" computers. Besides, nothing much may come of this and all it can do is cause embarrassment to IBM, because, in my opinion, we don't belong in the personal computer business to begin with.[13]

The two most important decisions made by Frank Cary, IBM's chairman and CEO, were not only that the development of an IBM personal computer or PC should go ahead, but also that its development could proceed outside of IBM's normal processes. In particular, the Boca Raton team was free to build the system using a non-IBM microprocessor, and they chose to use the new 16-bit Intel 8088 chip. An 8-bit microprocessor, as used in the first generation of personal computers, could only access 8 bits of data in a single machine instruction. The next generation microprocessors like Intel's 8088 could access and process 16 bits at a time. In a further significant break from IBM's standard practices, Lowe also had permission to outsource the software to vendor companies. A 1979 business study undertaken for IBM evaluating the prospects for microprocessor-based computers had advised the company not to develop proprietary systems and applications because "in order to succeed IBM would need a lot of third parties writing software for the new system."[14] The conclusion was clear: IBM would purchase an operating system from an outside company. This decision also implied that the vendor of the operating system could put its software on non-IBM machines.

B.8.6. Don Estridge (1937–85) led Project Chess – the top-secret project to develop an IBM PC at its Boca Raton plant in Florida. In an unprecedented move by IBM, the machine had an open architecture and used third-party hardware and software. Estridge died in a plane crash three years after the PC's introduction – by then the PC was a runaway success and IBM had sold more than a million machines.

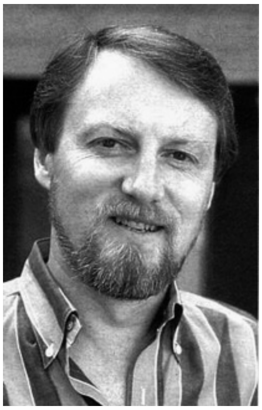
Jack Sams was the IBM engineer in charge of software development for the PC prototype. In the summer of 1980, Sams led a delegation from IBM to Microsoft's offices in Seattle, where they briefed Allen and Gates about their top-secret effort to build an IBM personal computer, code-named Project Chess. According to IBM historian Edward Bride:

> Sams met with Bill Gates to evaluate whether Microsoft could handle the task of writing a BASIC compiler for the IBM PC. This led to his recommendation to William Lowe that they use Microsoft software in the final product. In addition, when he was unable to make a deal with Intergalactic Digital Research for the operating system, Sams and his team turned to Microsoft. This led to the development of an operating system released by IBM as PC-DOS and by Microsoft as MS-DOS.[15]

Microsoft agreed to supply compilers not only for BASIC but also for FORTRAN, COBOL, and Pascal, all delivered on IBM's tight timetable.

Intergalactic Digital Research, later shortened to Digital Research, was a company set up by Gary Kildall (B.8.7) to market his CP/M software, an operating system for microprocessors, including personal computers. CP/M was then the leading *disk operating system* (DOS) for computers with one or more disk drives. *Dr. Dobb's Journal*, a magazine aimed at computer programmers, had announced CP/M to hobbyists in 1976 as being similar to DECSYSTEM 10 in that it used commands derived from DEC's operating system software. For example, it specified a disk drive by a letter; file names had a period and a three-character extension; and the "DIR" command enabled the user to see the available files in a directory. In 1977, Kildall had rewritten CP/M so that only a small part of the software needed to be customized for each new machine. He called this specialized code the BIOS, for Basic Input/Output System. The BIOS standardized personal computer system software in the same way that the Altair bus had standardized the hardware.

For reasons that are still unclear, the IBM delegation decided they could not reach an agreement with Digital Research for CP/M and came back to Microsoft. Their return presented Microsoft with a dilemma because the company was not at that time in the business of writing operating system software. Concerned that the whole deal with IBM might now be in jeopardy, Allen and Gates looked around for alternatives. A local company called Seattle Computer Products (SCP) was producing 8086 16-bit hardware, and a designer from SCP, Tim Paterson, had been working with Paul Allen on testing his prototype hardware using Allen's 8086 BASIC software. As an interim measure while waiting for Gary Kildall to deliver his long-promised 16-bit version of CP/M, Paterson had also developed a program he called QDOS, standing for Quick and Dirty Operating System. Paul Allen and Gates then made a deal with Rod Brock, the owner of SCP, for Microsoft to license QDOS, now renamed 86-DOS. In July 1981, Microsoft went back to Brock and negotiated the outright purchase of all rights to 86-DOS. This deal was probably the best value in the history of computing and provided the foundation for Microsoft's future success.



B.8.7. Gary Kildall (1942–94) had a PhD in computer science from the University of Washington and was teaching at the Naval Postgraduate School in Monterey, California, when he developed the first commercially successful operating system for microcomputers – Control Program for Microcomputers or CP/M in 1974. He and his wife then established a company – Digital Research – to market CP/M. IBM approached Kildall about providing CP/M for its PC project but for reasons that remain obscure, Kildall and IBM were unable to reach an agreement. IBM then went back to Microsoft, who then created their phenomenally successful PC-DOS operating system.

The IBM Boca Raton team had committed to delivering a hardware prototype to Microsoft "before December 1" of 1980. It was actually delivered early in the morning of Monday, 1 December. Microsoft's business manager, Steve Ballmer, answered the door and showed the IBM team to a small, windowless backroom, which was kept under lock and key, with access limited to only a handful of people. Difficulties with the unreliable hardware caused problems for Microsoft's software teams, and they missed the original mid-January deadline for both PC-DOS and BASIC. In the end, the IBM PC, IBM's personal computer, was announced in August 1981 and shipped ahead of schedule in November (Fig. 8.12). Besides PC-DOS from Microsoft, there were two other operating systems available – CP/M-86 from Digital Research and p-System from the University of California, San Diego. As David Bradley said, "Simple economics determined the winner – PC-DOS sold for about $40, while CP/M-86 and p-System were about $400."[16] The IBM planners had estimated that "in the five-year lifetime of the IBM PC, sales from all sources would equal 241,683 units."[17] The corporate staff at IBM actually scolded the planners for suggesting such unrealistically large sales volumes. In fact, according to Bradley: "Over the

Fig. 8.12. The IBM PC. IBM's first personal computer arrived in 1981, more than five years after personal microcomputers first arrived. However, the IBM name instantly legitimized the business market and gave companies the confidence to invest in personal computers for word processing and spreadsheet work. Although IBM had introduced the PC in 1981 with an advertising campaign aimed at the general public, the IBM PC had its most profound impact in the corporate world. Companies bought PCs in bulk, revolutionizing the role of computers in the office – and introducing MS-DOS to a vast user community. Unlike most previous IBM products, the PC incorporated hardware and software from other companies. The PC also had an open architecture, which allowed a thriving "PC clone" business to develop.

Fig. 8.13. An IBM PC button featuring a Charlie Chaplin–like figure.

PC's five-year lifetime, IBM sold approximately 3 million systems, 250,000 in one month alone in 1984."[18]

During the next few years, the IBM PC became an industry standard, and most of the popular software packages were converted to run on the machine (Fig. 8.13). In January 1983, the editors of *Time* magazine nominated the IBM PC as their "Man of the Year." The openness of the architecture and the standardization of the operating system software encouraged other manufacturers to produce *IBM-compatible computers*, also called *IBM clones*, which copied the features of the IBM PC. IBM remained the technology leader and produced several very successful successors to the original PC, most notably its second-generation personal computer, the PC AT – the letters AT stood for *advanced technology*. The PC AT bus allowed expansion by the easy insertion of printed circuit boards. In 1987 IBM tried to introduce some proprietary technology into the PC market with the Personal System/2 or PS/2 computer, replacing the now-standard but limited 16-bit PC AT bus with the more capable Micro Channel Architecture. Although IBM was willing to license the technology to others, the strategy to regain a proprietary advantage was not a success. Eventually the PC AT bus was superseded by the Peripheral Component Interconnect (PCI) interface, an architecture created by an industry consortium in 1993. As Mark Dean, a participant in the original IBM PC design team, now says:

> I'd have to admit that we lost sight of why the PC had become successful when we went to the PS/2. To enable continued growth, we should have continued with the model of building it so that other people can play. That would have allowed us to stay in control of the market. When we did the PS/2, we lost control.[19]

The Macintosh and Microsoft Windows

For all of the creativity at Xerox PARC in the 1970s, the success of personal computing – first with the Apple II and then with the IBM PC – owed nothing to any of their research. This situation changed in 1979 when Steve Jobs was invited to visit PARC. At the insistence of Xerox higher management, PARC showed Jobs its Alto-based vision of the office of the future. Larry Tesler remembers Jobs asking, "Why isn't Xerox marketing this? … You could blow everybody away!"[20] In fact, microprocessor technology was not yet powerful enough to support all the features he had seen. When Xerox released the Xerox Star in 1981, it was not a commercial success despite wonderful reviews and its many advanced features, such as the capability to network the computer to a

Fig. 8.14. The Macintosh computer was announced in 1984 in a now-famous advertisement during the U.S. Super Bowl football game. The video was made by Ridley Scott and contrasted the regimented world of IBM's PC dominance with the creativity made possible by the Macintosh with explicit reference to Big Brother and George Orwell's novel *1984*.

laser printer and other Star machines. The Star was too expensive (more than $10,000) to compete with the "good enough" approach of the IBM PC released later that year. As a result of his visit to PARC, Jobs recruited Larry Tesler to lead the development of a new Apple computer to be called the Lisa, named after Jobs's daughter. The Lisa was launched in 1983, but, at a price point of nearly $10,000 like the Star, it was not a commercial success. However, there was another new Apple computer in the works.

The Macintosh project had been started in mid-1979 by Jef Raskin, who had been a professor of computer science at the University of California in San Diego. He was familiar with the work at PARC, and he wanted to produce a machine with a built-in screen that was so simple and easy to use that a user could just plug it in and get started right away. The machine was called the Macintosh, after Raskin's favorite apple. When Jobs returned from PARC, he took over the project. Raskin had wanted to produce a machine for less than $1,000. At Jobs's insistence, Apple added new PARC-like features including a mouse and this increased the price. The Macintosh finally went on sale in 1984 for nearly $2,500 (Fig. 8.14). To build the hardware and the software for the Macintosh, Jobs isolated the design group in a separate building over which a pirate's flag was hoisted. John Sculley, later CEO of Apple, said:

> Steve's "pirates" were a hand-picked band of the most brilliant mavericks inside and outside Apple. Their mission, as one would boldly describe it, was

B.8.8. Steve Jobs (1955–2011) was a university dropout who played a key role in shaping today's computing universe. With the talented engineer, Steve Wozniak, Jobs founded Apple Computer in 1976 to market the Apple I personal computer kit. The Apple II was released in 1977 as a self-contained consumer product that was great for playing games but also ran the VisiCalc spreadsheet software, the first killer application for business. After a famous visit to Xerox PARC in 1979 at which Jobs saw the Alto personal computer and its GUI, Apple produced the revolutionary Macintosh computer in 1984.

After falling out with the Apple Board and CEO John Sculley, Jobs was effectively fired from Apple in 1985 and sold all but one of his shares. He then founded the NeXT computer company and its first computer workstation was released in 1990 – and used by Tim Berners-Lee, at CERN in Geneva, to develop the World Wide Web. NeXT reported its first profit of just more than $1 million in 1994.

In 1986, Jobs bought a 70 percent stake in a graphics company later called Pixar that helped Disney computerize its ageing animation department. Pixar's digital animation business was originally just a sideline to their hardware and software business. Jobs was losing money at both NeXT and Pixar but all this changed in 1995 with the success of Pixar's full-length animated movie *Toy Story*, with Jobs credited as executive producer.

In 1996, Apple had lost market share dramatically, and Jobs was invited back to Apple as an adviser with an agreement that Apple would buy NeXT for around $400 million. By 1997, Jobs had the title interim CEO, inevitably abbreviated as iCEO. In the first year that Jobs came back, he laid off more than three thousand employees and Apple lost more than $1 billion in 1997. After two years of huge losses, Apple had returned a $300 million profit by 1998. As CEO, Jobs oversaw a succession of phenomenal successes – starting with the iMac, followed by the iTunes store, the iPod, the iPhone, and the iPad. With its touch interface, Jobs completely reinvented the mobile phone as can be seen at a glance by the number of people using touch phones in every situation. In 2003, Jobs was diagnosed with cancer of the pancreas and although an initial treatment had some success, his health declined and he died in October 2011.

to blow people's minds and overturn standards. United by the Zen slogan "The journey is the reward," the pirates ransacked the company for ideas, parts, and design plans.[21]

Fig. 8.15. IBM personal computer advertisement. Marketing computers to students was a new experience for IBM. They had certainly never before suggested to customers that they could use them "under your favorite tree."

Unlike the Lisa, which achieved excellent performance by using more expensive specialized hardware, the Macintosh attempted to provide Lisa-like features using the commodity Motorola 68000 microprocessor. Jobs obsessively oversaw every aspect of the Macintosh's design, even including the colors of the production facilities and the designers of the case, and his name appears on the design patent. One of the designers, Terry Oyama, later said: "Even though Steve didn't draw any of the lines, his ideas and inspiration made the design what it is. To be honest, we didn't know what it meant for a computer to be 'friendly' until Steve told us."[22] Jobs got each of the forty-seven people from the original Macintosh design team to sign their names inside the molding: these original Macs are now collectors' items.

In spite of the Mac's impressive capabilities, it was not successful as a consumer product, and the lack of an option to incorporate a hard disk meant that it could not displace IBM from the business market (Fig. 8.15). It did gain a loyal following in the publishing and media industries, where it came to the fore because of its powerful capabilities for *desktop publishing*, in which editors and designers used computers to edit text and lay out pages. Unlike the Apple II and the IBM PC, the Macintosh architecture was closed, and third parties were not able to add circuit boards offering additional functionality. Although Microsoft supplied some application software, Apple had developed its own operating system and it was difficult for developers to write applications that made optimal use of the hardware.

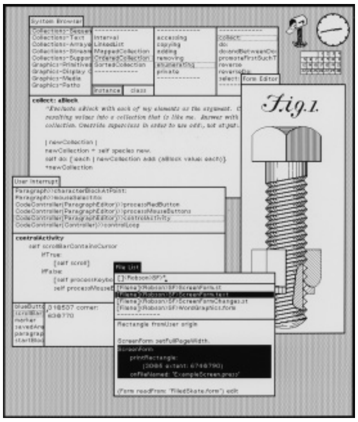


Fig. 8.16. Screenshot of the Cedar environment running on a Xerox Alto. The GUI had windows, icons, menus, and a pointing device – the WIMP interface we still use today. Steve Jobs was inspired by his glimpse of the Alto's GUI that he saw at Xerox PARC. Jobs was sure that this was the way forward for personal computing and he committed Apple to this approach.

Users liked the exciting look and feel of the Macintosh's *graphical user interface* – abbreviated as GUI and pronounced "gooey." This was the revolutionary system developed by researchers at Xerox PARC that enabled users to give instructions to the computer through a *WIMP interface* – standing for windows, icons, menus and pointers. The use of *windows* here is not a reference to Microsoft's operating system of the same name, but to a rectangular frame called a *window* that appears on the computer screen. A window can run a program at the same time as other windows on the same screen are running other programs. The user can see the output from all the programs on the screen and can enter information into any program by selecting the corresponding window. *Icons* are small pictures representing specific actions that the user can select. A *menu* is a list of available options, typically shown by icons and a *drop-down menu*, which lists programs or applications when selected by the user. Lastly, a *pointer* is a marker, such as an arrow, that appears on the screen to allow the user to select an operation. For a long time the most common way of controlling the pointer has been with a *mouse*, a palm-sized device that enabled the user to move an arrow on the screen and to select icons of drop-down menus by clicking a button.

The success of the Mac GUI made it clear that the next important step would be the development of a similarly powerful interface for the IBM PC and its clones (Fig. 8.16). Several companies, including Digital Research and IBM, attempted to produce a similar interface for the PC. Microsoft had begun work on a GUI project after Bill Gates had visited Jobs at Apple and had seen the prototype Macintosh computer in development. The Microsoft product was originally going to be called "Interface Manager," but Scott McGregor, who had joined Microsoft from PARC, had written the window manager component for PARC's interactive programming environment and had called his PARC software "Windows." Rowland Hanson, the head of marketing, then persuaded Gates to call Microsoft's new operating system "Windows," as Hanson explained, "to have our name basically define the generic."[23] Version 1 of Windows appeared in 1985. The highest performance Intel microprocessor for the PC at the time was the 80286, called the "286" for short, but even on this chip the Windows GUI ran far too slowly. It was only when the Intel 386 and 486 chips became available in the late 1980s that using windows on Windows really became a practical proposition. Meanwhile, the company had also developed a new GUI-based operating system called OS/2 with IBM, released in 1987. But by early 1989, Microsoft had sold some two million copies of Windows and OS/2 was history. When Windows 3.0 launched in May 1990, Bill Gates (B.8.10) was finally able to say that it "puts the 'personal' back into millions of MS-DOS-based computers."[24] However, it was not until the release of Windows 3.1 in 1992 that the original PARC vision of computing for the masses truly arrived.

During the 1980s, Microsoft had been developing application software for the Macintosh and, in so doing, had learned how to develop software for a windows-based interface. When Microsoft designers applied this experience to the PC, Gates followed the example of Steve Jobs in insisting that each application adhere to a common GUI. With Charles Simonyi (B.8.9) having left Xerox PARC and now at Microsoft developing Word for Windows, and with the Excel spreadsheet program, first developed for the Mac, Microsoft could finally put these together with the PowerPoint presentation software to form an "office suite." By a wholehearted commitment to GUIs, and by bundling three applications together as Office, Microsoft was finally able to overtake the PC market leaders for word processing and spreadsheets, WordPerfect and Lotus 1-2-3.

Meanwhile, Microsoft's lawyers were battling a lawsuit filed by Apple charging that Microsoft had infringed "the Company's registered audio-visual copyrights protecting the Macintosh user interface."[25] After four years of legal

B.8.9. Charles Simonyi wrote Bravo, the first WYSIWYG word processor, while at Xerox PARC. He later "took the PARC virus" to Microsoft where he was responsible for creating the hugely successful Word for Windows application. Simonyi also helped develop a system of programming that allowed Microsoft to manage increasingly complex software projects involving large teams of programmers. The style involved a systematic way of naming variables – called "Hungarian" because of its apparent incomprehensibility. Simonyi has used some of his personal fortune from his time at Microsoft to become an astronaut – as shown here – and he has visited the International Space Station on two occasions.

B.8.10. Bill Gates is one of the best-known faces of the personal computer revolution and it would be hard to find a person who would not recognize his name. At the age of thirteen, he was enrolled in Lakeside School, an exclusive preparatory school in Seattle. When Gates was in eighth grade the school purchased an ASR-33 teletype and some computer time for students on a GE computer and he wrote his first BASIC programs. With Paul Allen and some other Lakeside students, he was allowed free use of a DEC PDP-10 computer at the nearby offices of the Computer Center Corporation provided they assisted in debugging the operating system software. At age seventeen, Gates and Allen formed their first joint venture called Traf-O-Data for making hardware and software for automating a traffic counting system. The enterprise was not a success but Gates and Allen developed valuable experience and a powerful set of tools for the PDP-10. Gates graduated from Lakeside in 1973 and enrolled at Harvard. Publication of the January 1975 issue of *Popular Electronics* stimulated Allen and Gates to develop a basic interpreter for the Altair 8800 computer. In 1976 they established Microsoft Corporation to develop software for the growing microcomputer market. Microsoft's partnership with IBM to develop the MS-DOS operating system for the IBM PC was a critical step for the company. Bill Gates had a remarkable vision for Microsoft: "a personal computer on every desk and in every home." With the advent of the Internet and the World Wide Web, in 1995 Gates turned the company around to embrace the Web with his famous "The Internet is a tidal wave" memo. In 2006 Bill Gates transitioned out of his day-to-day involvement with Microsoft and now devotes a significant amount of his time to philanthropic activities with the Bill and Melinda Gates Foundation (B.8.11). With fellow billionaire Warren Buffett, Gates champions the cause of "creative capitalism" – a combination of capitalism and philanthropy to solve some of the urgent problems facing the world.

arguments, a federal judge dismissed Apple's lawsuit in 1992, ruling that "Apple cannot get patent-like protection for the idea of a graphical user interface, or the idea of a desktop metaphor [under copyright law]. . . ."[26] In Walter Isaacson's biography of Steve Jobs, Bill Gates is quoted as ending an angry meeting with Jobs by saying: "Well, Steve, I think there's more than one way of looking at it. I think it's more like we both had this rich neighbor named Xerox and I broke into his house to steal the TV set and found out that you had already stolen it."[27]

A post-PC era?

With the progress of Moore's law, the scale of computers has been extended from large "mainframe" business computers to microprocessor-based personal computers. From Osborne's first portable computer – which was more "luggable" than truly portable – we now have smart phones and tablets that are rapidly changing the way we interact with computers. These are more than just new "form factors" for the PC where the term *form factor* refers to the size,

B.8.11. Melinda and Bill Gates visiting with mothers taking part in a malaria intervention treatment program at the Manhiça Health Research Center in Mozambique. Bill and Melinda announced in Manhiça that their foundation was awarding three grants totaling $168 million to fight malaria. The grants will accelerate the search for a malaria vaccine, new drugs to fight drug-resistant malaria, and new treatment strategies for children.

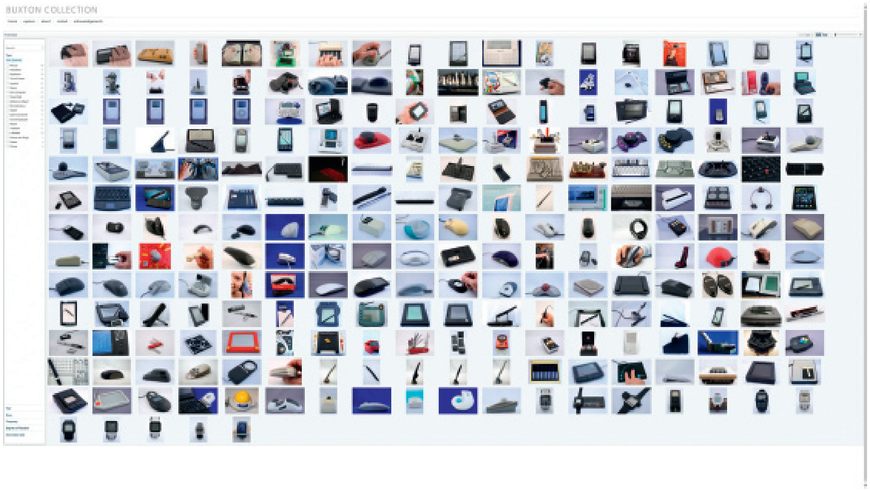


Fig. 8.17. Bill Buxton is a Canadian computer scientist and designer and a pioneer in the research field of human-computer interactions. The Buxton Collection is a collection of interactive devices that he has been collecting for about thirty-five years. The collection is intended as a resource for those interested in design, user experience, and the history of interaction. Buxton is a principal researcher at Microsoft Research and an expert on birch bark canoes.

configuration, and physical arrangement of a hardware device. Because these devices can now be with us all the time new modes of interaction are threatening to displace the mouse as our primary interaction mechanism (Fig. 8.17).

The term *smart phone* was introduced by Ericsson in 1997. A smart phone is just a mobile phone that uses a microprocessor-based computing platform to provide the computing power we expect from a PC. The first smart phone was the "IBM Simon" introduced in 1993. In addition to being a mobile phone it served as a *personal digital assistant*, providing a calendar, address book, calculator, notepad, and clock. Simon ran a version of DOS and could also play games but only a few thousand were sold. In 2002, the Canadian telecommunications company Research in Motion (RIM) introduced its first BlackBerry smart phone, which combined the ability to send and receive email with the capabilities of a mobile phone. We will explore the emergence of the Internet, email, and the World Wide Web in Chapters 10 and 11. Easy access to email and the web, together with the increasing availability of "Wi-Fi" allowing wireless connectivity to the Internet, have been two of the key drivers for the emergence of new portable computing and communication devices such as smart phones and tablet computers.

Many companies have tried to market smart phones, tablets, and personal digital assistants with a variety of different user interfaces. The history of touch screen input goes back a long way. The first touch screen using capacitive technology was invented by E. A. Johnson at the Royal Radar Establishment in Malvern, United Kingdom, in the 1960s for an air-traffic control application. The device works by sensing the change in electric charge caused by a finger touching the screen. Another common type of touch technology used on Point of Sale systems is based on the change in resistance caused by pressing on a flexible surface. Other mechanisms for interacting with computers include

Fig. 8.18. The first “Newton” message pad from Apple was released in 1993. This handheld device used an ARM RISC processor and ran various applications for handwriting recognition, note taking, sketching, Internet access, and other productivity tools. The more sophisticated iPhone incorporates many features of its predecessor.

voice input, handwriting with a stylus, and gesture recognition. However, it was not until Steve Jobs unveiled the iPhone in 2007 that there has been mass adoption of such devices by consumers (Fig. 8.18). Instead of a mouse or a stylus, the iPhone used touch as its input mechanism and transformed the user experience.

Jobs and Apple also created an “App Store” where third-party vendors could market their applications for the iPhone. By 2012, more than half a million applications were available for download – copied electronically to the user’s own iPhone. In terms of usage, in the first two months of 2012, more than half of user sessions on iPhones were spent playing games! From being a frivolous application wasting valuable computer time, Moore’s law has transformed the economics so that computer games are now seen as immensely valuable. It is this theme that we take up in Chapter 9.

Key concepts

- Bus
- 16-bit microprocessor
- Graphical user interface
- Desktop metaphor
- WIMP Interface: windows, icons, menus, and pointers
- Mouse
- Bitmap
- Touch screen input

“... AND MR. WILSON WANTS TO KNOW IF YOU’VE SAVED YOUR DRAWING”

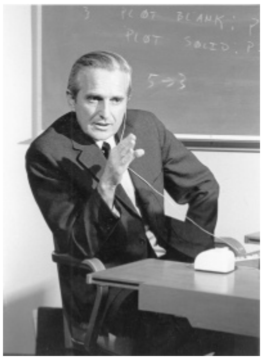
Three pioneers of interactive computing

Licklider and human-computer symbiosis

Although his name is not well known to the general public, few people have been more influential in the evolution of computing than J. C. R. Licklider (see Chapter 10, B.10.1, for a brief biography). Lick, as he asked everyone to call him, studied psychology as an undergraduate at Washington University in St. Louis and followed up with a PhD in psychoacoustics at the University of Rochester. After a spell at Harvard, Lick moved to MIT in 1950. His interest in information technology and human-computer interactions had led him to become involved with the Semi-Automated Ground Environment (SAGE) air-defense system. The Cold War threat of an airborne nuclear attack on the United States had triggered the development of SAGE. The system derived from Jay Forrester's original Whirlwind project at MIT. Forrester's team built a prototype for the SAGE system, and IBM, the lead contractor for the project, produced the system for installation. Each SAGE computer could track up to four hundred airplanes and support up to fifty terminals. From his experience with this project, Lick had become convinced of the value of computers to analyze data in real time, as the data came in, instead of being confined to the traditional batch processing of mainframe computers. Lick summarized his ideas in an influential paper called "Man-Computer Symbiosis," published in 1960, arguing for the need to develop computers that could interact with humans to support real-time decisions – that is, decisions that need to be made at the actual time that events were happening. In 1962, Lick was given a unique opportunity to implement his vision for the future of computing. He was invited to lead a new research program at the U.S. Department of Defense's Advanced Research Projects Agency (ARPA). Lick was based in the Pentagon and had, as Michael Hiltzik says, "one 'cloak-and-dagger' project under his nominal jurisdiction [that was] so highly classified even he was not cleared to know what it was."[28] Licklider's Information Processing Techniques Office (IPTO) ultimately had a larger computing research budget than all of the other U.S. government agencies combined. Lick's strategy was to place his trust in a small number of talented individuals and outstanding centers of academic excellence. Lick nurtured interactive computing research not only at MIT but also at the University of California in Berkeley, Carnegie Mellon University, Stanford University, and the University of Utah. He gave researchers significant amounts of funding that enabled them to pursue long-term research goals without too much interference or frequent proposal writing. His program of interactive computing ultimately delivered major advances in many important areas, including networking, computer graphics, software engineering, and human-computer interactions. A major component of his program was a $3 million grant to MIT for Project MAC (Project on Mathematics and Computation), a pioneering time-sharing system that eventually could support up to thirty users at any one time. Lick's ARPA centers at Utah and Stanford generated almost all of the ideas embodied in today's computer user interfaces. David Evans and Ivan Sutherland headed the graphics research group of at the University of Utah, and Doug Engelbart (see B.8.12) led the Human Factors Research Center at the Stanford Research Institute.

One of Lick's most significant contributions while at ARPA was his role in helping establish computer science as a valid research discipline in universities. Bob Taylor, one of his successors at ARPA, said:

> Prior to his [Lick's] work at ARPA, no U.S. university granted a Ph.D. in computer science. A university graduate program requires a research base, and that in turn requires a long-term commitment of dollars. Lick's



B.8.12. Doug Engelbart (1925–2013) was best known as the inventor of the mouse. In fact Engelbart played a leading role in many major developments in computing. This photo shows him rehearsing for the demo in 1968, which has entered computing history as the "Mother of All Demos."

ARPA program set the precedent for providing the research base at four of the first universities to establish graduate programs in computer science: U.C. Berkeley, CMU, MIT, and Stanford. These programs, started in 1965, have remained the country's strongest and have served as role models for other departments that followed. Their success would have been impossible without the foundation put in place by Lick in 1962–64.[29]

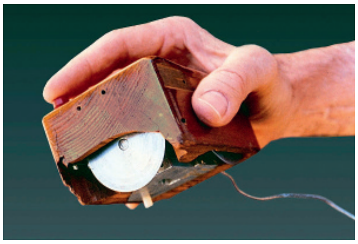


Fig. 8.19. Doug Engelbart's mouse from 1967. This prototype mouse, invented by Engelbart at the Stanford Research Institute, rolled on two sharp wheels facing 90 degrees from each other.

Doug Engelbart and the mouse

On Engelbart's return home from military service in World War II, he had been inspired by reading Vannevar Bush's visionary essay "As We May Think," published in 1945. Bush accurately foresaw the days when scientists would be drowning in information:

> Publication has been extended far beyond our present ability to make real use of the record. The summation of human experience is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentarily important item is the same as was used in the days of square-rigged ships.[30]

Bush envisioned a machine he called the "memex": "a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility."[31] In 1957, when Engelbart joined the Stanford Research Institute (SRI), he was finally able to start realizing his dream of creating a memex with funding initially from Bob Taylor, then at NASA, and later from Licklider at ARPA. (See Chapter 10, B.10.10 for a brief biography of Taylor.)

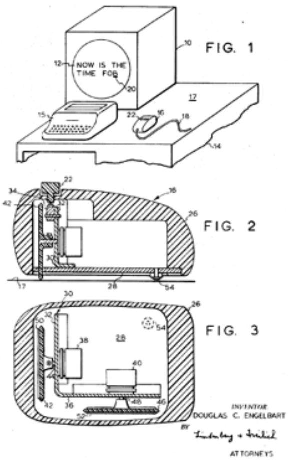


Fig. 8.20. Engelbart's "mouse patent" drawings. The word *mouse* does not appear in Engelbart's patent for the computer pointing device. The knife-edged wheels each rolled in just one direction, transmitting movement information for that direction. Each slid without turning when the mouse was moved in the other direction.

Engelbart and his team are best known for their invention of the mouse (Fig. 8.19), but they also pioneered many other features of the present-day GUI, in which the user controls a cursor on the screen to select options from menus, start programs by clicking icons, and perform other operations. Engelbart was not sure why it was called a mouse: "None of us would have thought that the name would have stayed with it out into the world, but the thing that none of us would have believed either was how long it would take for it to find its way out there."[32]

Engelbart's researcher, Bill English, created the first mouse out of a hollowed-out block of wood with two small wheels that allowed the user to control the movement of a cursor on the computer screen (Fig. 8.20). At an event that has been called the "Mother of All Demos" at a major computing conference in San Francisco in December 1968, Engelbart demonstrated his group's "electronic office" software, called NLS (short for oN Line System), in which he introduced the mouse, video conferencing, word processing, a real-time editor, and split-screen displays to the world. He also demonstrated a prototype of Bush's memex idea by showing how the user could select a single word in a text document and be instantly linked to a second document. This prototype was the first implementation of a *hypertext* system, which enables the user to jump from one document to another, such as we now use daily on the World Wide Web. Butler Lampson and Peter Deutsch, both early recruits to Xerox PARC, had worked part-time for Engelbart in the 1960s and were both influenced by the vision of the NLS electronic office software and by the 1968 demo.

Bob Taylor and Xerox PARC

Bob Taylor (see B.10.10) had a master's degree in psychology and was working for NASA as a project manager when he was invited to the Pentagon to meet Licklider. The two scientists had a shared background in psychoacoustics but also shared a vision for the future of interactive computing. When Licklider left ARPA in 1964, he persuaded Ivan Sutherland, creator of Sketchpad, one of the first interactive graphics programs, to leave MIT and take on management of the IPTO program with Taylor as his deputy. Sutherland stayed only a short time so Taylor soon found himself running the entire IPTO program. He continued to support the embryonic U.S. computer science community and a vision of interactive and networked computing. He organized annual IPTO research conferences and gained an unrivaled personal knowledge and the trust of the most creative individuals in the U.S. computing research community. This served him well when he was appointed to recruit researchers for the Computer Science Laboratory at Xerox's brand new Palo Alto Research Center (PARC) that had just opened in June 1970.

Fig. 8.21. Lab Director Bob Taylor held periodic informal meetings in the "bean-bag" conference room where his Xerox PARC staff presented their new technical ideas. Speakers always received frank and honest feedback from their colleagues.

PARC was the inspiration of Xerox's CEO, Peter McColough, who realized that the copier market would soon become much more competitive with the expiration of one of Xerox's key patents. McColough wanted Xerox to own what he called the "office of the future." The new mission for Xerox was to control "the architecture of information." Taylor brought together a cast of computing superstars at PARC. These stars included Butler Lampson and Chuck Thacker, both rescued from a failed Berkeley start-up called the Berkeley Computer Corporation, and Alan Kay, one of Ivan Sutherland's research students from the University of Utah. Kay's vision was to build a "Dynabook" – "a notebook-shaped machine with a display screen and a keyboard you could use to create, edit, and store a very personal sort of literature, music, and art,"[33] – exactly the sort of vision that Taylor and Licklider had wanted their IPTO program to generate. One more element of the mix at PARC came from another of their investments, Engelbart's Augmentation Research Center at the nearby Stanford Research Institute.

Taylor wanted to incorporate elements of Engelbart's vision of interactive computing into PARC, so he recruited Bill English, the engineer who had done the detailed design work for the mouse. Eventually others from Engelbart's team followed, and the ideas of the Augmentation Research Center team and their NLS electronic office system became essential elements of PARC's own vision for interactive computing. At PARC, the stage was set. Taylor believed that, having provided the researchers with the overall vision and funding, his job was now to keep out of the way and let them do what they did best.

Taylor was a key player in the history of Xerox PARC (Fig. 8.21) but was also a controversial figure. Nevertheless, in his resignation speech he could fairly say:

> Most people spend a lifetime without opportunities for pioneering completely new ways of thinking about large collections of ideas. I have been fortunate to have been a leader in three: time-sharing; long-distance interactive networking; and personal distributed computing.[34]

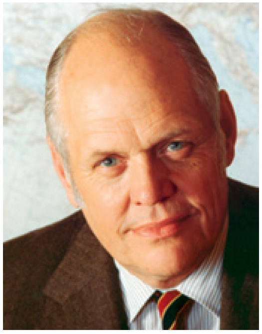
From minicomputers to portable computers

The rise and fall of DEC

Although users had been able to run their own programs on early computers like the ENIAC and the EDSAC, a new usage model developed as the market for commercial computers exploded in the 1950s. A new occupation called *computer operator* emerged. Computers sat in air-conditioned machine rooms, and computer operators kept the machines running and loaded jobs onto the computer instead of allowing users direct access to the hardware. This system was called *batch processing*, and although it made efficient use of the very expensive computer, it could also be extremely frustrating for users trying to debug their programs. Typically, a program consisted of a stack of cards on which the user had punched holes to represent data. The computer operators fed the cards into the computer, and the user received a printout for the program some hours later, or even the next day if the program was run overnight. As you can imagine, users found it extremely tiresome to wait twelve hours for their program to run. Sometimes, they picked up the printout only to find an error message saying that the program could not run because the computer had received an incorrect instruction, which might be as minor as a mistyped comma in one of the program statements. As we saw in Chapter 3, such frustration led John McCarthy to develop the idea of time sharing, in which multiple users could be connected to the computer simultaneously and have the illusion that they were the sole user. The computer switched its attention from user to user, executing a small part of each user's program during each slice of time. Because even the early computers could perform many thousands of operations each second, users had the impression that their program was running all the time. Users interacted with the machine using their own computer *terminal*, typically just a combination of keyboard and screen, but they could also input programs and data using much faster paper tape or punch card readers.

The efforts of McCarthy and others at MIT achieved success in 1961 when Fernando Corbató introduced the Compatible Time-Sharing System (CTSS), one of the first working time-sharing systems, running on the MIT Computation Center's IBM computers. IBM was skeptical at first, but the success of time sharing in the early 1960s led to the establishment of new companies that offered commercial time-sharing services. Customers would buy time on the expensive machines and pay for it by the minute. For a few brief years, time sharing seemed like the path to the future, but it was soon overtaken by the development of the minicomputer. In 1957, Ken Olsen (B.8.13), an electrical engineer from MIT, had the bold idea of starting a new type of computer company. Olsen had worked with Jay Forrester on Project Whirlwind to develop a computer for the U.S. Navy. In 1952, while still a graduate student, Olsen and his fellow graduate student Harlan Anderson had played a major role in building a machine called the Memory Test Computer to try out Forrester's ideas on magnetic core memories, working with one of the MIT computing pioneers, Wesley Clark.

What was Olsen's bold idea? Frustrated with the slow development of interactive and time-sharing computing at MIT, he decided that



B.8.13. Ken Olsen (1926–2011) began his career by fixing radios in his basement in Bridgeport, Connecticut. As a graduate student at MIT, he and fellow student Harlan Anderson built the Memory Test Computer to evaluate the feasibility of using magnetic core memory. While at MIT's Lincoln Labs, they were responsible, with Wesley Clark, for designing and building the first transistorized research computer, the TX-0. With his brother and Harlan Anderson, Olsen founded the first minicomputer company, the Digital Equipment Corporation. Their first computer, the PDP-1 was based on the TX-0. From 1957 until 1992, Digital's headquarters was located in a former wool mill in Maynard, Massachusetts.

there was a market for small, inexpensive machines he called *minicomputers*. Many of the computational problems required by the business and the research community were actually relatively small, such as calculating a payroll or monitoring an experiment. So in 1957, Olsen, his colleague Harlan Anderson, and Ken's younger brother Stan decided to go into the computer business for themselves. With several thousand dollars of their own money, supplemented by funds from a Boston investment firm, they set up the Digital Equipment Corporation – also known as DEC and later just as Digital – in a Civil War–era wool mill just outside Boston, Massachusetts. Three years later, they produced their first computer, the Programmed Data Processor model 1, commonly known as the PDP-1. This machine cost $120,000 and provided much more cost-effective computing than was then available from IBM and others. DEC's business really took off with the introduction of the PDP-8 in 1965, generally regarded as the first minicomputer (Fig. 8.22). The PDP-8 machine used transistors and magnetic core memory and cost $18,000. It could only run one program at a time and had less memory than a mainframe computer, but it became the first commercially successful minicomputer. The key selling point was its price and ability to be easily coupled with laboratory instruments for experimentation and control. Because of the low cost of the PDP-8, many more customers could afford to buy their own computer to do their routine computational tasks. As computer historian Stan Augarten reports in his 1984 book *Bit by Bit:*

Fig. 8.22. A PDP-8 on a tractor used for controlling sowing.

> Scientists ordered PDP-8s for their laboratories; engineers got them for their offices; the Navy installed them on submarines. In refineries, PDP-8s controlled the flow of chemicals; in factories, they operated the machine tools; in warehouses, they kept track of inventory; in computing centers, they ran programs that didn't require the power of a mainframe; in banks, they kept track of accounts. The notion of the information utility gave way to *distributed processing*. For example, a bank would install a minicomputer in each of its branches; the machines handled the branches' transactions during the day and sent records of their transactions to the bank's central computer at closing time. The applications were endless.[35]

While Bob Metcalf was a graduate student at MIT and Harvard, DEC lent him a PDP-8. It was stolen from his lab and he had no idea how he could repay DEC. However, the company took the news in its stride and ran an advertisement for the PDP-8 as "the first computer small enough to steal."[36]

In 1965 DEC pursued a second path for low-cost, interactive computing by introducing the PDP-6 as the first commercial time-sharing system. This used many of the concepts and functions of MIT's CTSS software and also DEC's experience with a time-shared PDP-1 at BBN, specially designed for Licklider. Introduced in 1965 and the forerunner of the PDP-10, the PDP-6 was a 36-bit, time-shared mainframe computer with roughly the same power as the IBM 709X and 110X series mainframe batch computers. Thus DEC grew rapidly, beginning in the mid-1960s and through the 1970s along two paths: classical minicomputers like the PDP-8 and PDP-11 (introduced in 1970), and the PDP-10 time-shared computers that could support one hundred or more active users and were used by universities and by time-sharing service companies. The PDP-10 ran a time-sharing system called TOPS-10. The Computer Center Corporation or "C-Cubed" installed one of the first PDP-10s in the Seattle area in 1968. To help debug the system, the company offered free time on the computer to a couple of local teenagers named Paul Allen and Bill Gates. In the early 1980s, the two paths were covered by the PDP-11 minicomputers that used single chip microprocessors, and the VAX-11 computers that were typically time shared and could be used in clusters. By 1980, almost one hundred companies had started building minicomputers using integrated circuits. By 1985, only six of these companies remained.

The architecture of the PDP-11 popularized the idea of a *bus*, a set of connections linking all the major components of the machine, including the central processor, memory, and I/O devices, in a standard way. The bus architecture was important because it allowed both DEC and "original equipment manufacturers" – often known as OEMs – to easily add extra units and to customize the machine for specialized applications. By the mid-1970s, the minicomputer market had become very competitive and DEC needed to offer a computer with more memory than the PDP-11, a 16-bit computer that could access 64 kilobytes of memory. The VAX 11/780, announced in October 1977, was the first commercially available 32-bit computer. It supported 2^{32} or 4 gigabytes of virtual address space. *Virtual memory* is a mechanism for swapping data in and out of a small, fast main memory from a slower, larger memory on a *hard disk*, a rigid magnetic disk permanently mounted in the computer's disk drive and used to store data. The VAX had sixteen 32-bit registers and could understand a large, complex *instruction set* (set of commands). The computer architect Gordon Bell (B.8.14) led the initial design effort for the VAX, as head of DEC's R&D organization. Bell had served as the architect of many of DEC's successful machines. The VAX turned out to be a runaway success. It offered cost-effective high performance compared to the much more expensive mainframe computers. It also had a user-friendly operating system called VMS and came with a standard set of languages and library software (Fig. 8.23).

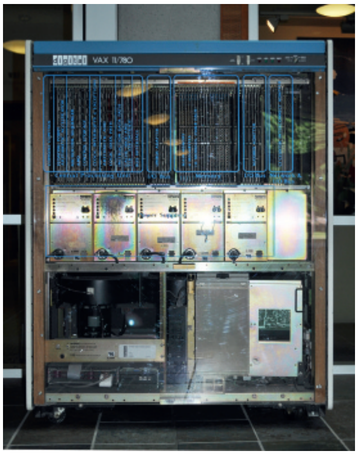


Fig. 8.23. DEC's very successful VAX-11/780 computer was the result of a small architecture team of six engineers led by Gordon Bell, including Dave Cutler, who was responsible for the VMS software architecture and its implementation. The VMS operating system was much liked by users but universities often put up a version of the UNIX operating system as an alternative. Both Bell and Cutler later went on to work for Microsoft.

Despite these early successes that allowed DEC to become the second-largest computer company through the 1980s, DEC no longer exists. Although it built the foundation for inexpensive interactive computing, it missed out on the personal computer revolution. The IBM PC was announced in 1981 and used Intel's 8088 microprocessor, which had all the essential features of a computer on a single chip. So why was DEC not able to succeed in this new market? Although DEC's founder, Ken Olsen, is often quoted as saying, "There is no reason for any individual to have a computer in his home,"[37] DEC really tried hard to succeed with personal computers. In 1982, DEC introduced three incompatible personal computers – the DECmate, based

B.8.14. Gordon Bell grew up in Missouri and helped in the family electrical business, repairing appliances and wiring homes. He graduated from MIT with a degree in electrical engineering and then spent time in Australia programming an English Electric DEUCE computer, a production version of Alan Turing's Pilot ACE computer. In 1960, Bell was recruited by DEC where he worked on the early PDP machines and designed the first UART – Universal Asynchronous Receiver/Transmitter – that converted bytes of data and transmitted the individual bits sequentially. As a faculty member at Carnegie Mellon University, with Allen Newell, he introduced the processor-memory-switch (PMS) and instruction-set processor (ISP) notations for describing computer structure and architecture. When back at DEC in the 1970s as the head of their R&D organization, he led the design team that developed the enormously successful VAX computer. In the 1980s, Bell was the founding Assistant Director of the U.S. National Science Foundation's Computer and Information Science and Engineering Directorate. In 1997 he established the Gordon Bell Prize for outstanding achievement in high performance computing applications. In 1995 he joined Microsoft where he built a version of the memex of Vannevar Bush. His recent work on the MyLifeBits project aims to capture digitally all the significant events each day in a person's life including geographical locations, conversations, phone calls, messages sent, and even the web pages visited.

on the PDP-8 and sold as a word processing machine; the DEC Professional, a more powerful machine than the IBM PC but based on the PDP-11 architecture and a proprietary bus; and the DEC Rainbow, an "almost IBM PC compatible" platform. DEC engineers prided themselves on their expertise in computer architecture and "refused to be part of the pack and compete with others by supplying competitive but fully compatible machines."[38]

Nevertheless, it is too simplistic to attribute DEC's demise solely to its failure in the PC market. Although it is true that DEC had made several bad management decisions, in the 1990s, with the rise of the Internet and the web (see Chapters 10 and 11), the company was still well placed to become a market leader for Internet products. DEC had extensive expertise in networking and servers, and also had pioneered one of the first successful web search engines with their AltaVista offering. In the end, Gordon Bell, Vice President of Engineering at DEC during the 1970s, believes that "Failure was simply ignorance and incompetence on the part of DEC's top 3–5 leaders and, to some degree, its ineffective board of directors that in removing Olsen made an even worse mistake in appointing [his replacement] Palmer."[39]

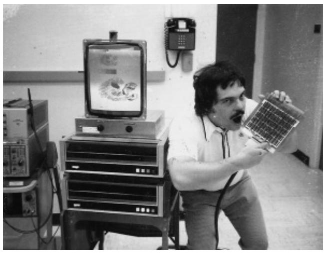
The time machine: The Alto

In 1972, Chuck Thacker (B.8.15), Butler Lampson (B.8.16), and Alan Kay (B.8.17) at Xerox PARC conceived of building a revolutionary new type of computer. Instead of batch processing or time sharing on a mainframe or minicomputer, the "Alto" was intended to be a genuine personal computer small enough to fit under a desk (Fig. 8.24). To computer designers and businesses at the time, computers were expensive devices. Just to provide the computer memory for a single-user machine would cost many thousands of dollars. "But to Thacker and his colleagues such objections missed the point," Hiltzik says, and explains:

B.8.15. Charles (Chuck) Thacker (left) and Butler Lampson at Xerox PARC. Thacker is a Turing Award recipient and designer of Xerox PARC's Alto computer – the first truly personal computer. He had learned from his experience at the unsuccessful Berkeley Computer Corporation that in designing computing systems, "less is often better than more." Thacker's word for describing engineering projects that had got out of hand was "biggerism"– as in "This project has been biggered." Thacker also had an influence in introducing the WYSIWYG capability in the Bravo word processor. When his wife Karen was typing a paper for a class, he suggested she try using an early version of the Bravo word processor. She commented that she needed to see what she got in print on the screen. Thacker passed this comment to his colleagues at PARC and Bravo was soon able to do this.

B.8.16. Butler Lampson was the software architect of Xerox PARC's famous Alto personal computing system. At PARC, he also made major contributions to the first WYSIWYG word processor, Ethernet for local area networking, operating systems, and laser printers. Lampson received the ACM Turing Award in 1992 and the remarkable citation read: "For contributions to the development of distributed, personal computing environments and the technology for their implementation: workstations, networks, operating systems, programming systems, displays, security, and document publishing."[F4] His wife, Lois Lampson, was the first person to produce her PhD thesis on a laser printer. When she submitted her thesis, the administrator insisted on knowing which was the original – to be deposited in the library – and which was the copy! This photo was taken at the Rome NATO software engineering conference in 1969.

Fig. 8.24. The Xerox PARC Alto, which featured a mouse, removable data storage, networking hardware, a visual user interface, easy-to-use graphics software, and email. The Bravo and Gypsy word-processing software offered the user the first "What-You-See-Is-What-You-Get" or WYSIWYG interface, with printed documents matching what users saw on screen. The Alto for the first time combined these and other now-familiar elements in one small computer. Developed by Xerox as a research system, the Alto marked a radical leap in the evolution of how people interact with computers, leading the way to the environments we still have on today's computers. By making human-computer interactions more intuitive and user friendly, the Alto opened computing to a much wider range of users, from experts to nonspecialists, including children. People were able to focus on using the computer as a tool to accomplish a task rather than on learning their computer's technical details. When it was built, the revolutionary Alto would have been a very expensive personal computer if put on sale commercially. Lead engineer Charles Thacker noted that the first Alto probably cost in the region of $12,000 to build: as a product, the price tag might have been as much as $40,000. A decade later, Moore's law had reduced costs and personal computers with adequate memory became affordable.



B.8.17. Alan Kay's name is closely linked with the development of personal computing. He started college but left before graduation to join the air force. In the air force he found a new interest in computing and when he left he enrolled at the University of Colorado. Kay graduated in 1966 with degrees in mathematics and molecular biology and went on to graduate work at the University of Utah where he obtained an MS in electrical engineering and a PhD in computer science in 1969. It was at Utah that Kay conceived of the *Dynabook* – a portable, personal computer rather like the iPad of today, but he could not create it with the technology of the time. He joined Xerox PARC in 1971 and his research team created the overlapping windowing GUI interface. Kay was also one of the creators of the Smalltalk programming language and coined the name *Object Oriented Programming*. The picture on the screen is the first animated bitmapped graphic: the Sesame Street cookie monster eating a cookie. Kay received the 2003 Turing Award for pioneering object-oriented programming and "for fundamental contributions to personal computing."

> The Alto aimed to be not a machine of its time, but of the future. Computer memory was horrifically expensive at the moment, true, but it was getting cheaper every week. At the rate prices were falling, the same memory that cost ten grand in 1973 would be available in 1983 for thirty dollars. The governing principle of PARC was that the place existed to give their employer that ten-year head start on the future. They even contrived a shorthand phrase to explain the concept. The Alto, they said, was a time machine.[40]

Thacker, Lampson, and Kay all agreed that they needed to build a fast, compact machine with a *high-resolution* display – that is, a display with images that were sharp and finely detailed. For Alan Kay, it would not be a complete realization of his vision but it would at least be, as he said, an "interim Dynabook."[41] Thacker began designing the machine in November 1972, and the first prototype was up and running in an incredibly short time by April 1973. One of the major challenges was powering a high-resolution display without using unreasonable amounts of processor power and memory. Thacker's solution was to use a *bitmap*, a representation of an image consisting of rows and columns of dots. Each bit in the computer's memory corresponds to a dot or *pixel* on the display screen. This bitmap had been inspired by experiments in Kay's group that used a block of memory that normally stored custom fonts to display images. The screen had a resolution of 606 by 808 pixels. This meant that nearly half a million bits needed to be refreshed thirty times a second, which was a great deal of processing power and memory for the time. As processing power and computer memory became cheaper, these limitations rapidly disappeared, as predicted by Lampson. The Alto was the future.

Despite the excitement generated by the Alto, Lampson was only too well aware that it still needed real application software to be useful. Lampson was sketching the requirements for a text-editing program when Charles Simonyi (B.8.9) walked into his office. Simonyi had been an undergraduate at Berkeley when

Lampson was a graduate student and they had collaborated on the CAL TSS time-sharing system. He had also worked with Lampson at the ill-fated Berkeley Computer Corporation while still a student. After working for a while on a parallel computing project called Illiac-IV, he rejoined his ex-Berkeley colleagues at PARC. Lampson supplied Simonyi with three sheets of notes capturing his thoughts for an interactive text editor. Simonyi called his new word processing system "Bravo." Using the Alto's bitmapped screen, he was able to encode complex typefaces, boldface, italic, and underlining in the text together with a detailed page layout, so that the document appeared on the screen almost exactly as it would be printed out. Bravo was thus the first WSYIWYG word processor – What You See Is What You Get – and it became a great hit among the engineers at PARC. As Simonyi later said:

> It was the killer app, no question. People would come into PARC at night to write all kinds of stuff, sending letters, doing all personal correspondence, PTA reports, silly little newsletters, anything. If you went around and looked at what the Altos were doing, they were all in Bravo.[42]

In spite of its popularity with the engineers at PARC, Bravo needed a more user-friendly interface if it was to be adopted by the much larger community of nonengineers. Lampson and Thacker had made a deliberate decision not to work on the user interface of Bravo, not because they did not think it was important but because they did not have the resources to do both the implementation and the user interface. It was left to Bob Taylor to initiate such a project with two other computer scientists at PARC, Larry Tesler and Tim Mott.

Before joining PARC, Tesler had produced a program called Pub that helped ordinary users format and print their documents. At PARC, Tesler had been a member of a team trying to reengineer and update a version of Engelbart's interactive multimedia system. He rapidly became dissatisfied with the complexity of the system being created and was eager to take on a new challenge. Tim Mott was an Englishman with a computer science degree from the University of Manchester who was working in the United States for a Xerox subsidiary called Ginn & Company that published textbooks. Determined to try to get some value from Xerox's "corporate research" tax, Mott's boss Darwin Newton sent Mott to visit PARC and see how their research on office systems could assist him as a publisher. Mott concluded that their system was much too complex and difficult for the publishing company to use: "There wasn't a lot of time spent looking at what mere mortals would be able to do with the system."[43] Taylor challenged Mott to use the Alto to produce something useful.

Tesler and Mott also found the user interface of Bravo far too complicated. It was usable by experts but not easy or attractive for ordinary users like publishers. Mott went back to Ginn & Company and did some market research on what nonengineers actually wanted from such a program. Unsurprisingly, he found that the publishers wanted the program to mimic what they actually did with their paper-based process. This is the origin of the "cut" and "paste" commands that are still used to this day. Tesler and Mott called their new system Gypsy, and it was the first program to use the mouse to execute point-and-click operations in the way we do today.

While Simonyi, Tesler, and Mott were developing Bravo and Gypsy, Alan Kay's group at PARC was still pursuing his Dynabook dream. The Alto's bitmapped screen allowed enormous flexibility in what could be displayed. So why can't a user write a memo in one part of the screen and use a drawing program in another part? This led the team to think of the screen in terms of a "desktop" metaphor where electronic documents could be piled on top of one another, just like papers on a desktop. They created overlapping boxes, or "windows," for each different task. But shifting these boxes around put a huge demand on the Alto's processor and was extremely slow. In a stroke of genius, Dan Ingalls came up with "BitBlt," an abbreviation of "bit boundary block transfer" and pronounced *bitblit*. Instead of having the computer change each of the components of a rectangular image individually for the new location, BitBlt operated on the entire bitmap using fast Boolean operations to create the new image. This new technique meant that the user could rapidly scroll up or down the text of a document on the screen by moving a mouse. It also meant that windows could be created and moved around at will, and that the illusion of a stack of papers on a desk could be convincingly executed. When Kay's team was demonstrating their system to their skeptical engineering colleagues, Ingalls stunned the audience by using a mouse click to call up a drop-down menu listing several possible commands and selecting the "cut" command. As Hiltzik says:

> The PARC user interface, with its overlapping windows, mouse clicks, and pop-up menus, had entered computing history. More than twenty-five years and many engineering generations later, it remains the indisputable parent of the desktop metaphor guiding the users of millions of home and office computers.[44]

Bob Taylor's leadership was the key ingredient in PARC's astonishing success. According to Butler Lampson:

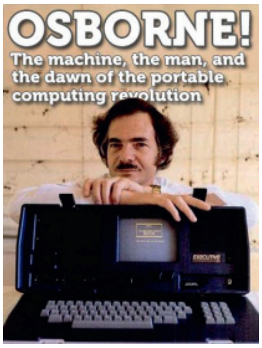
> The master often speaks in somewhat inscrutable fashion with a deeper and more profound interpretation than his humble disciples are able to provide. In retrospect you can really see that the path has been plotted years in advance, and you've been following his footsteps all along.[45]

Chuck Thacker agreed: "As a leader of engineers and scientists he had no equal. If you're looking for the magic, it was him."[46]

The Osborne portable computer

At the beginning of the 1980s, many small companies were entering the personal computer market with creative ideas. In July 1981, a British computer designer named Adam Osborne (B.8.18) launched a portable personal computer that became a big success with traveling business executives. One of the advantages was that it was designed like a briefcase that could fit under the airplane seat. The computer was based on the popular Z80 microprocessor designed by Federico Faggin, who had designed the first Intel microprocessor back in 1971. The Osborne 1 (Fig. 8.25) had two floppy disk drives; 64 kilobytes of memory; a five-inch, fifty-two-column scrollable display; and a modem connection that could send and receive data by telephone. The reason for the small display was portability: a larger display could be easily damaged during transportation. The Osborne 1 ran the CP/M operating system and its bundled software included a BASIC interpreter, the Word-Star word processing software, and a SuperCalc spreadsheet program. The price of the computer was very attractive and generated a huge demand. The Osborne Computer Corporation grew from two employees to three thousand within a year. However, Osborne Computer made some critical mistakes that caused its sudden decline in a fiercely competitive market. The company declared bankruptcy in 1983. Adam Osborne described the reasons for this demise in his book *Hypergrowth: The Rise and Fall of Osborne Computer Corporation*.

Fig. 8.25. The first "Portable Computer" – the Osborne 1 – was released in 1981. It weighed 24.5 pounds (12 kg) and cost US$1795 – just more than half the cost of a computer from other manufacturers with comparable features – and ran the CP/M operating system. It was designed to fit under an airline seat. At its peak, the Osborne Computer Corporation was shipping ten thousand units per month.



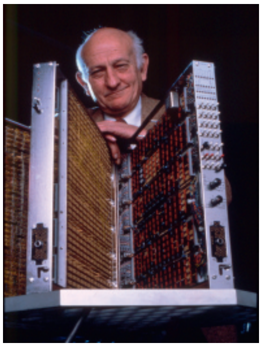
B.8.18. Adam Osborne (1939–2003) was best known for creating the first commercially available portable computer, the Osborne 1, released in April 1981. The Osborne 1 included both word processing and spreadsheet software and the company was briefly very successful. The Osborne 1 was produced at a time when IBM did not bundle hardware and software with their PCs: customers had to buy the operating system software and the monitor separately.

Developments in computer architecture: RISC and ARM

The idea of RISC – standing for Reduced Instruction Set Computing – originated on the East and West coasts of the United States at around the same time in the early 1980s. At IBM Research in the 1970s and 1980s, John Cocke (B.8.19) had investigated how often the individual instructions of an instruction set were actually executed when running a representative set of programs. He discovered that a small set of instructions occurs more frequently than others and proposed that only this reduced instruction set should be implemented in hardware. The more complex instructions of the standard approach can then be built up out of this smaller set. Having only a small instruction set simplifies the circuit design and enables us to build fast computers with low power consumption.

On the West Coast, David Patterson at Berkeley and John Hennessy at Stanford were pursuing similar ideas. It was Patterson who coined the name RISC, for *reduced instruction set computing* architecture – in contrast to the usual *complex instruction set computing* (CISC) architecture of a standard microprocessor. The first RISC processor was introduced in an experimental research computer called the IBM 801 in 1980. Cocke's ideas made their way into the IBM POWER architecture – an acronym for Performance Optimization With Enhanced RISC. This led to the introduction of IBM's RS/6000 (RS for RISC System) workstations in 1990. Cocke's colleague, Fran Allen, worked with him on the interaction of computer architectures and compilers and they were responsible for developing many innovative compiler optimization techniques (see B.8.20). In recent years there has been a coming together of RISC and CISC. The new microprocessors of Intel's x86 series externally support a CISC instruction set of almost nine hundred instructions, but internally only a RISC subset of instructions are actually implemented in silicon.

For smart phones and tablets, power consumption and battery life is very important. The United Kingdom–based company ARM Holdings – ARM standing for Advanced RISC Machines – had its origins in the Acorn computer company. In the United Kingdom, Acorn had great success with a personal computer called the BBC Microcomputer (Fig. 8.26). In looking for a microprocessor for their next generation machine, they took the unusual step of deciding to design their own microprocessor. Herman Hauser, the CEO of Acorn, encouraged two of his engineers, Steve Furber and Sophie Wilson, to look at the Berkeley RISC papers and then sent them on a fact finding visit to the United States. They visited Bill Mensch, CEO of the Western Design Center in Phoenix, Arizona, and were amazed at the tiny scale of his globally successful operation. As Wilson tells it: "A couple of senior engineers, and a bunch of college kids … were designing this thing. . . .



B.8.19. John Cocke (1925–2002) received a BS degree in mechanical engineering from Duke University in 1946 and later went back to Duke to complete a PhD in mathematics in 1956. He then joined IBM Research where he remained for the rest of his working life. At a symposium in honor of John Cocke in 1990, Fred Brooks described him as a "fire starter" because of his constant stream of ideas: "The metaphor that comes to mind is of a man running through a forest with flint and steel, striking sparks everywhere."[F5] After working on IBM's Stretch project, an ambitious effort to build the fastest scientific computer, and the Advanced Computer Systems research project, in 1975 Cocke led the research team building the experimental IBM 801 computer, which pioneered the ideas of RISC architectures and optimizing compiler technology. In the 1980s, these ideas led to the IBM POWER architecture and the RS/6000 RISC workstations. In 1987, Cocke received the Turing Award for the development of RISC and for his work on optimizing compilers with Fran Allen. In his 1990 talk, Fred Brooks characterized Shannon, von Neumann, and Aitken as the three "greats" of the first generation of computer scientists; and Knuth, Sutherland, and Cocke as the three "greats" of the next generation.

B.8.20. Fran Allen grew up on a farm in New York State and obtained an MSc degree in mathematics from the University of Michigan. She joined IBM in 1957 initially to earn enough money to pay off her school loans and stayed at IBM for the next forty-five years. Her first assignment was to teach IBM research scientists about the new FORTRAN language that had been developed by John Backus. This set her interest to compilers and she collaborated with John Cocke on the interaction of computer architecture and compilers in both the IBM Stretch and ACS projects. In 2006, Allen became the first woman to receive the Turing Award for her "pioneering contributions to the theory and practice of optimizing compiler techniques that laid the foundation for modern optimizing compilers and automatic parallel execution."[F7]

We left that building utterly convinced that designing processors was simple."[47] They came back to Cambridge thinking: "Well, if they can design a microprocessor, so can we."[48] Eighteen months later in April 1985, they had a working ARM chip.

ARM is different from other microprocessor companies such as Intel and AMD in that it licenses its technology to other companies rather than having its own manufacturing facilities. The strategy has been spectacularly successful. ARM technology is now used in 95 percent of smart phones and 80 percent of digital cameras and, by 2012, more than forty billion ARM-based chips had been shipped.

Fig. 8.26. The BBC Microcomputer. The British Broadcasting Company's Computer Literacy Project in the early 1980s hoped "to introduce interested adults to the world of computers."[F6] Acorn produced this popular computer in 1981 so viewers at home could emulate what they saw demonstrated on *The Computer Programme* TV series.

Bell's law for the birth and death of computer classes

In 1972, Gordon Bell observed that, with Moore's law predicting that the number of transistors would double every eighteen months, and with the introduction of Intel's 4004 microprocessor in 1971, it was possible to predict the broad outlines of the next forty years of computer evolution. He suggested that there would be two evolutionary paths for computers: (1) evolution at constant price with increasing performance; and (2) evolution at constant performance but with decreasing cost. Bell's law states that, roughly every decade, a new, lower-priced computer class forms based on a new programming platform that results in new usage patterns and the establishment of a new industry. Examples of such new classes are the emergence of the minicomputer, the personal computer, and the smart phone (Fig. 8.27).

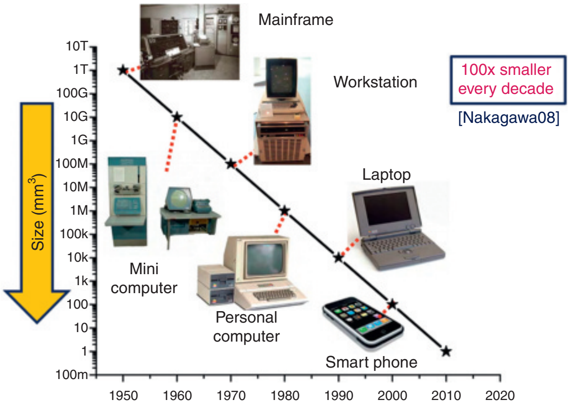


Fig. 8.27. A pictorial explanation of Bell's law.

Odds and ends

The origins of Control-Alt-Delete

Fig. 8.28. Ctrl-Alt-Del cushions for programmers to sleep better.

One of the most enduring features of personal computers that survives to this day was the brainchild of IBM engineer David Bradley. He was part of the original twelve-person IBM PC hardware design team and was responsible for developing the program to get the computer system started after you turn it on. The development team needed an easy way to reset and reboot the machine. The simplest solution would have been to include a reset button in the hardware, but it turned out that the mechanical construction of the PC made this method difficult. Bradley then decided to use the keyboard and deliberately chose a set of three keys that would be difficult to hit by accident: "Control" (Ctrl) and "Alt" were on the left side of the keyboard, while the "Delete" (Del) key was on the far right (Fig. 8.28). The Control-Alt-Delete combination for reset was originally intended only as a development tool, but the command soon entered into general use. Because IBM had a deadline and the reset command was just one of many problems to solve, Bradley said, "After 10 minutes of design, coding and testing [of Control-Alt-Delete], it was time to move on to the next problem."[49]

Did QDOS copy CP/M?

One question that is sometimes raised is how similar Paterson's "Quick and Dirty Operating System" QDOS was to CP/M. In fact, Paterson had not had access to the source code of CP/M, and he had written QDOS using the CP/M user's manual and Intel's documentation for the 8086 microprocessor. Like CP/M, QDOS used the DEC system commands "Type," "Rename," and "Erase," and also retained Kildall's idea of how to make it convenient to modify and customize the computer system by using a BIOS for input and output. One of the DEC commands in CP/M that was changed in QDOS and MS-DOS was the all-purpose *PIP* command, where PIP stands for Peripheral Interface Program, a method used for transferring files between devices. This command was replaced by the much less cryptic "Copy" instruction. QDOS also introduced a more efficient file system for storing floppies by using Microsoft's File Allocation Table or "FAT" file system, which is still widely used today.

Fig. 8.29. The sledgehammer is about to smash the screen with its "Big Brother" image in Apple's famous 1984 Super Bowl advertisement for the Macintosh. The commercial was directed by Ridley Scott and produced two years after the release of his dystopian science fiction movie *Blade Runner*. In 1995 it was ranked as one of the "50 best" television commercials ever made by the editors of the trade publication *Advertising Age*.

The Mac and Big Brother

The Macintosh was introduced to the world in a legendary Super Bowl television advertisement in January 1984 that subtly linked the dominance of the IBM PC to *1984*, George Orwell's novel of a dystopian future (Fig. 8.29). *Hard Drive*, a history of Microsoft by James Wallace and Jim Erickson, describes the ad:

> [The commercial] showed a roomful of gaunt, zombie-like workers with shaved heads, dressed in pajamas like those worn by concentration camp prisoners, watching a huge viewing screen as Big Brother intoned about the great accomplishments of the computer age. The scene was stark, in dull, gray tones. Suddenly, a tanned and beautiful young woman wearing bright red track clothes sprinted into the room and hurled a sledgehammer into the screen, which exploded into blackness. Then a message appeared: "On January 24, Apple Computer will introduce the Macintosh. And you'll see why 1984 won't be like 1984."[50]