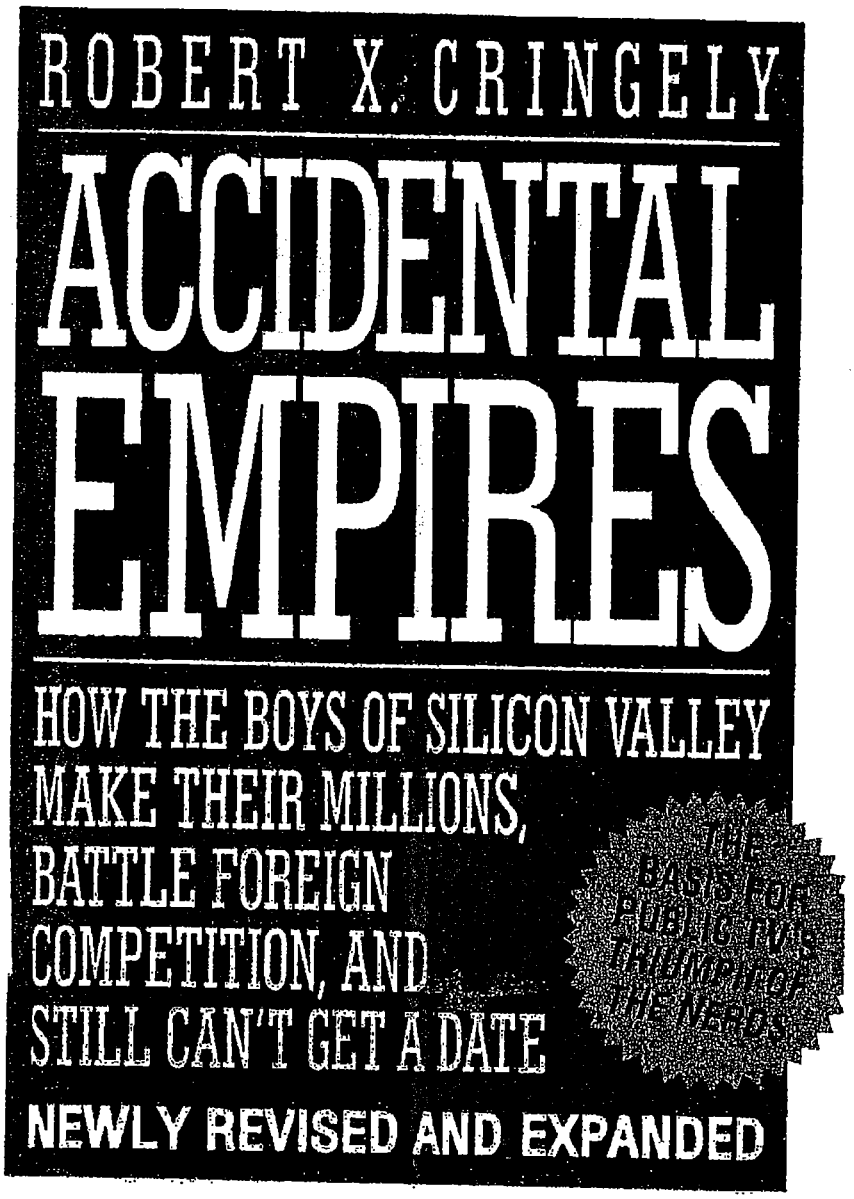
EXHIBIT AA

NATIONAL BESTSELLER



A hardcover edition of this book was published in 1992 by Addison-Wesley Publishing Company, Inc. It is here reprinted by arrangement with Addison-Wesley Publishing Company, Inc.

First HarperBusiness edition published 1993. Reissued 1996.

Designed by Janis Owens

The Library of Congress has catalogued the previous edition as follows:
Cringely, Robert X.
Accidental empires: how the boys of Silicon Valley make their millions, battle foreign competition, and still can't get a date / Robert X. Cringely.
p. cm.
Includes bibliographical references.
ISBN 0-88730-621-7
1. Computer industry—California—Santa Clara County. I. Title.
HD9696.C63U51586 1993
338.4'7004'0979473—dc20 92-53427

ISBN 0-88730-855-4
96 97 98 99 00 RRD 10 9 8 7 6 5 4 3 2 1

all of IBM's terminals so the new PC could be used to communicate with IBM minicomputers and mainframes. But that sales advantage was deliberately avoided because it would have killed the company's terminal business.

But wait. What about the operating system?

At the moment IBM was having its second meeting with Bill Gates, there were no 16-bit microcomputer operating systems on the market. If IBM had waited for Gary Kildall to get back from the airport, it might have learned that Digital Research was already working on CP/M-86, which would run on Intel's 16-bit 8086 microprocessor and on its little brother, the 8088. CP/M-86 would be ready to go about the time that IBM planned to release its personal computer, too, so it would have been a logical choice, had IBM known that CP/M-86 existed. As the largest seller of CP/M software, Microsoft knew that Digital Research had CP/M-86 coming down the chute, yet Gates, Allen, and Ballmer never mentioned it in their second meeting with IBM. Remember that the IBM nondisclosure agreement specifically urged them not to reveal any confidential information. CP/M-86 was clearly confidential.

Gary Kildall thought that he and Gates had divided the software market, with Digital Research taking the operating system business and Microsoft controlling the programming languages. Bill Gates knew better.

Across Lake Washington, at a company called Seattle Computer Products, was the operating system that Bill Gates wanted to sell to IBM. All he had to do was get it.

In business, as in comedy, timing is everything. There was nothing about QDOS, Seattle Computer Products' 16-bit operating system, that couldn't have been created just as well by programmers at Microsoft. But Microsoft programmers hadn't created it, and Tim Paterson of Seattle Computer Products had.

QDOS, which stood for "quick and dirty operating system," was a 16-bit clone of CP/M intended for an 8086-based computer being developed by the small company. All QDOS commands were the same as in CP/M. Paterson admitted to a little "low-level borrowing" from CP/M, too, but claimed that most of the code was his own.

Gary Kildall still thinks a lot of the QDOS code was stolen straight from his CP/M. "Ask Bill why function code 6 [in QDOS and still in MS-DOS, more than ten years later] ends in a dollar sign. No one in the world knows that but me."

There was nothing earthshaking about QDOS, except that it already existed. Bill Gates was buying time more than anything else when he paid Seattle Computer Products $50,000 for rights to the operating system. It must have seemed like a lot of money at the time.

Here's a great scene that never happened. Bill Gates flies to Florida, makes his pitch to IBM, offering to sell it a product called Quick & Dirty DOS, that, by the way, has at least some code stolen line for line from CP/M. The ears of Justice Department lawyers 900 miles away would have perked up. The IBM legal department, which was then suing Fujitsu for stealing IBM code, would have had a corporate seizure. And young Bill Gates would have found himself standing in the sun-drenched IBM parking lot wondering if it was something he said.

Instead, when Gates made that flight to Florida, he kept to the letter of IBM's own nondisclosure agreement and didn't reveal much about the true heritage of QDOS, now called MS-DOS, other than that it had been developed with the help of Seattle Computer Products.

Not everyone at Microsoft was so certain that the company ought to get into the operating system business. Microsoft was already running at full capacity just doing languages, so adding QDOS would require a major expansion. What if Microsoft

Warnock's garden in Los Altos, California. The new company defined the PostScript language and then began designing printer controllers that could interpret PostScript commands, rasterize the image, and direct a laser engine to print it on page. That's about the time that Steve Jobs came along.

The usual rule is that hardware has to exist before programmers will write software to run on it. There are a few exceptions to this rule, and one of these is PostScript, which is very advanced, very complex software that *still* doesn't run very fast on today's personal computers. PostScript was an order of magnitude more complex than most personal computer software of the mid-1980s. Tim Paterson's Quick and Dirty Operating System was written in less than six months. Jonathan Sachs did 1-2-3 in a year. Paul Allen and Bill Gates pulled together Microsoft BASIC in six weeks. Even Andy Hertzfeld put less than two years into writing the system software for Macintosh. But PostScript took twenty man-years to perfect. It was the most advanced software ever to run on a personal computer, and few microcomputers were up to the task.

The mainframe world, with its greater computing horsepower, might logically have embraced PostScript printers, so the fact that the personal computer was where PostScript made its mark is amazing, and is yet another testament to Steve Jobs's will.

The 128K Macintosh was a failure. It was an amazing design exercise that sat on a desk and did next to nothing, so not many people bought early Macs. The mood in Cupertino back in 1984 was gloomy. The Apple III, the Lisa, and now the Macintosh were all failures. The Apple II division was being ignored, the Lisa division was deliberately destroyed in a fit of Jobsian pique, and the Macintosh division was exhausted and depressed.

Apple had $250 million sunk in the ground before it started

INDEX

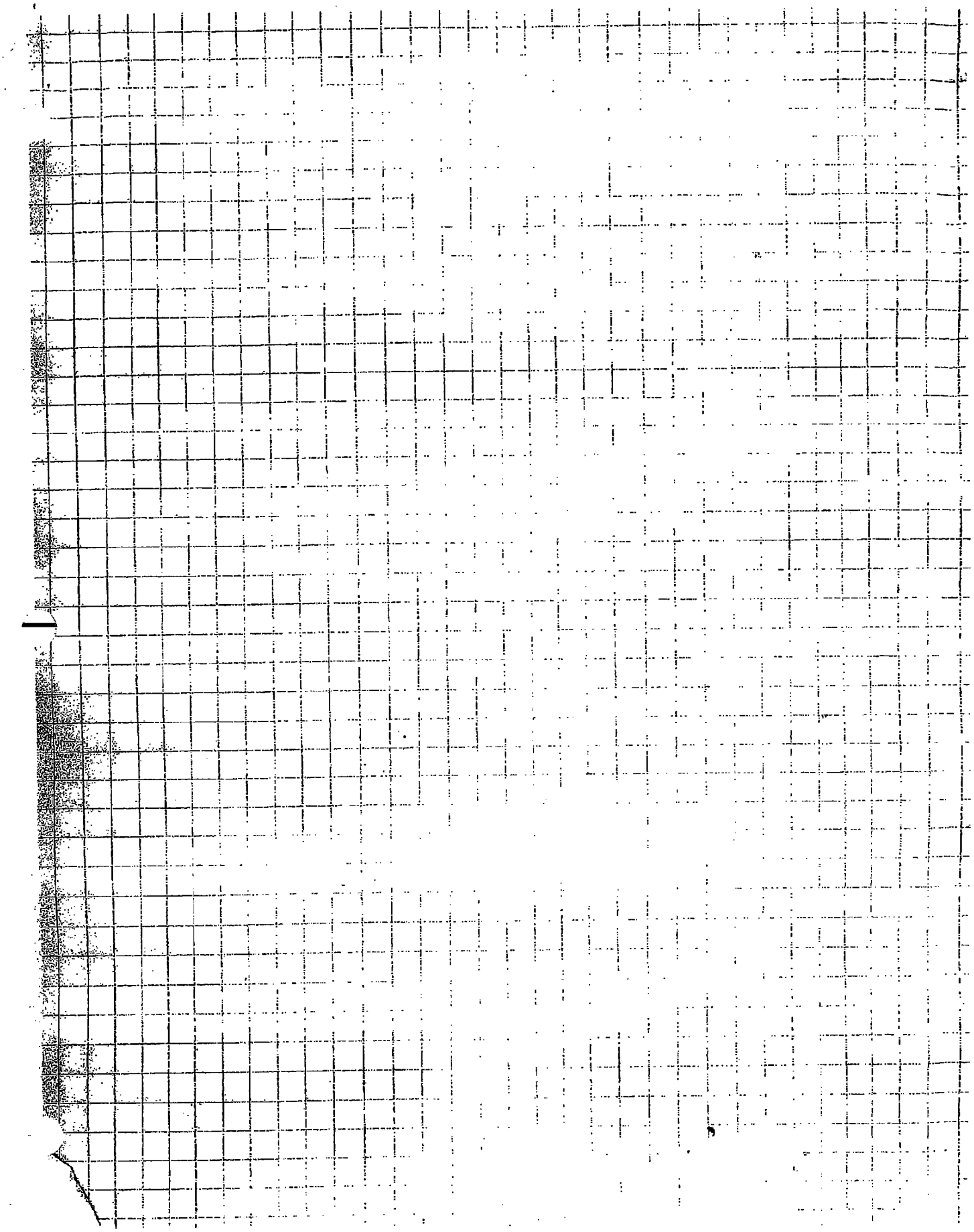
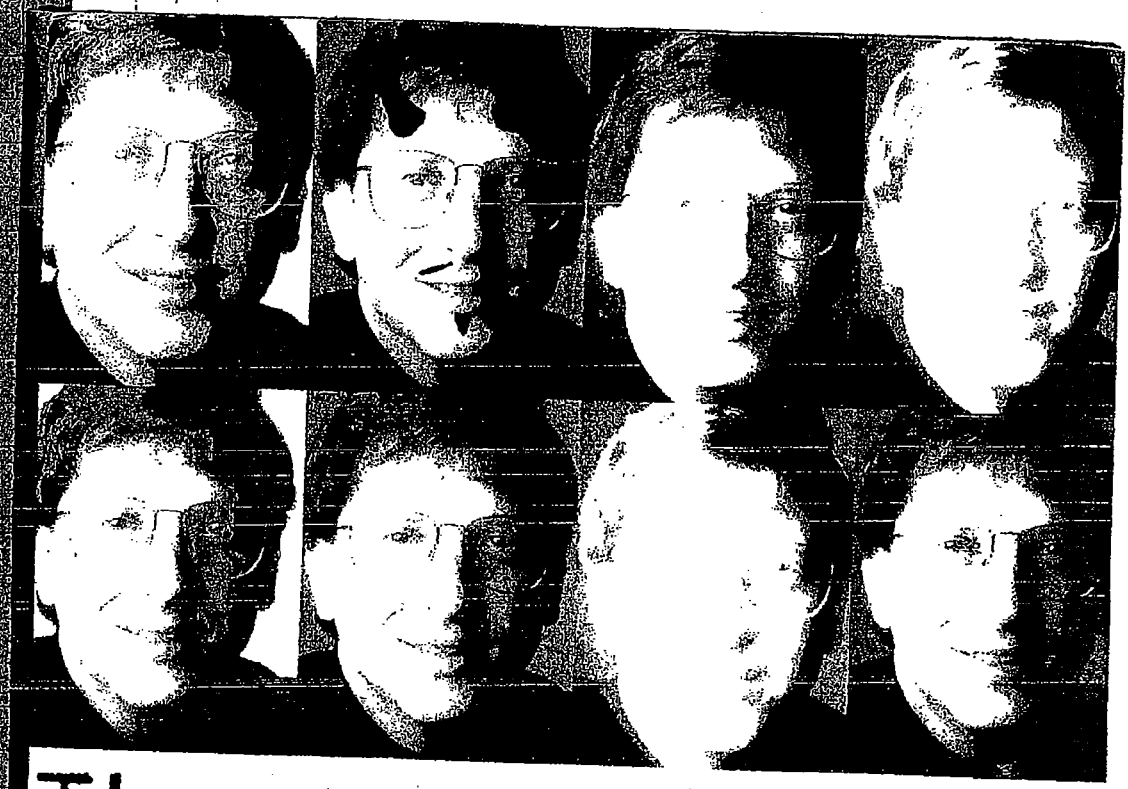


EXHIBIT BB



The to Get BILL GATES

An Irreverent Investigation of the World's Richest Man ... and the People Who Hate Him

Gary Rivlin

Paterson
EXHIBIT NO. 49
1-18-07
C. HAMMER

Library of Congress Cataloging-in-Publication Data

Rivlin, Gary.
The plot to get Bill Gates : an irreverent investigation of the world's
richest man . . . and the people who hate him / Gary Rivlin.
p. cm.
ISBN 0-8129-3006-1
1. Gates, Bill, 1955– . 2. Microsoft Corporation. 3. Computer software
industry—United States. 4. Competition—United States. I. Title.
HD9696.63.U64M5367 1999
338.7'610053'092—dc21
[B] 99-18228

Book Design by Robert C. Olsson
Random House website address: www.atrandom.com
Printed in the United States of America on acid-free paper.
2 4 6 8 9 7 5 3
First Edition

turn—and that no member of his team had been on any transcontinental flights with the Kildalls. Was an embarrassed Kildall masking the truth with fibs—or was Sams, who ended up taking a real liking to Gates, coaching and encouraging him like a son, the one telling tales? Whatever the truth, the day after his frustrating meeting at Digital Research, Sams flew back to Seattle. As Gates saw it, he had satisfied any obligation to Kildall when he made that first call, but since things had obviously not worked out, he was now free to do as he liked. So suddenly Microsoft was in the operating system business—and IBM was risking its top secret "Chess" project on the promises of twenty-four-year-old Bill Gates.

Time was everything to IBM, which was committed to getting its new PC to market in a year's time. As luck would have it, Paul Allen knew of a small company called Seattle Computer that sold what it called QDOS, for Quick and Dirty Operating System. The system was an obvious CP/M knockoff written only because its author, Tim Paterson, had grown frustrated waiting for Digital Research to update CP/M for the sixteen-bit systems that Seattle Computer sold. Paterson admitted that he had written QDOS with a CP/M manual at his side, intentionally mimicking key components to ease the task of developers accustomed to its popular predecessor (while at the same time improving on the original). The deal Microsoft hammered out with Seattle Computer gave it the right, for $25,000, to distribute QDOS to an unlimited number of users on behalf of an unspecified computer manufacturer. Microsoft dropped the Q and dubbed its new operating system DOS. Before the IBM deal, one of the few concessions Gates had made to a normal life was Sunday-night dinners with his parents. After inking the deal with Seattle Computer, however, he informed his mother that he probably wouldn't be showing up at any dinners for the next six months.

Somehow word leaked out about DOS, and suddenly computer manufacturers from around the world were phoning Microsoft headquarters. You could almost hear the *kerchung!* of the cash register going off in Gates's brain. It is said that there are two types of salesmen in the world, and they can be distinguished by their approach to selling a used car: those who would inform prospective buyers that a car had been in a serious accident and those who wouldn't. There was no doubting which kind Gates was. With all these companies interested in DOS, his goal was to buy all rights to QDOS before IBM unveiled its PC—before Seat-

tle Computer could appreciate the full worth of all it was sitting on. Seattle Computer offered to sell Microsoft all rights to its operating system for $150,000, but Ballmer turned up his nose at what he cast as "a grab for gold." The two sides settled on $50,000. In the end, Microsoft paid a total of $75,000 for a product that would earn the company billions in revenues and serve as the foundation of the rest of its successes.

"The PC came out in 1981," Nathan Myhrvold recalled. "By 1984, Digital Research would be a footnote."

THE BLOWN DEAL with IBM wasn't the end of Digital Research, though. Kildall began making noises about suing over the similarities between CP/M and DOS, so IBM invited him onto its machine. We'll give customers a choice, they told him. (An obviously displeased Gates would tell journalists that IBM had "blackmailed" him into agreeing, but for IBM, the importance of the DOS deal wasn't the magic of Microsoft but the insurance that it would be able to offer at least one workable operating system.) But Digital Research priced CP/M at $240 a copy, whereas the list price for DOS was $40. "Gary did everything he could not to compete," Myhrvold said with a sigh.

Factors other than price also hurt Kildall. An imitator enjoys the advantage of learning from a trailblazer's mistakes, and both Seattle Computer and Microsoft had improved on Kildall's invention. There was also Microsoft's favored status. The other software IBM sold for use on its PC was DOS compatible but not CP/M compatible. You could buy CP/M if you wanted to, but not only would you pay more, you'd have no guarantee that other programs would run on the machine.

Maybe Kildall's fate boiled down to a lack of ambition, or at least the kind of one-dimensional entrepreneurial ambition that has driven much of the innovation in the computer world. This man, described by a friend as "the biggest kid I've ever known," was passionate about a range of things; his computer business was only one toy among many. Gates was willing to do virtually anything a large customer asked; Kildall, by contrast, was a scientist protecting the purity of his invention from crass commercialism. Former Digital Research employees said that Kildall had no hunger to take his thirty-employee company, housed in a charming Victorian house, and turn it into a major company that played on Wall Street. Nor did he possess the desire to hire a hard-charging CEO, for fear an outsider would spoil the family feel of

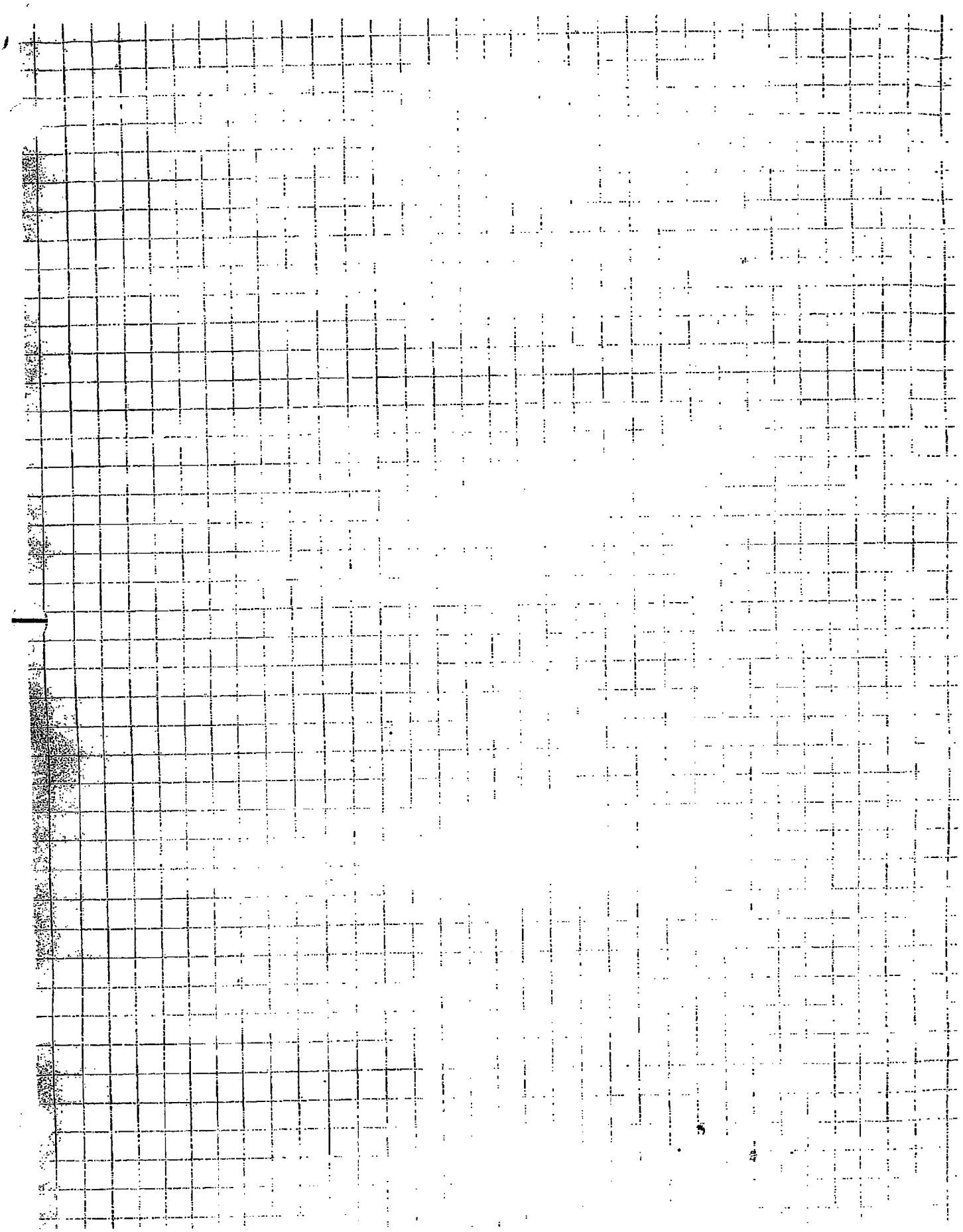
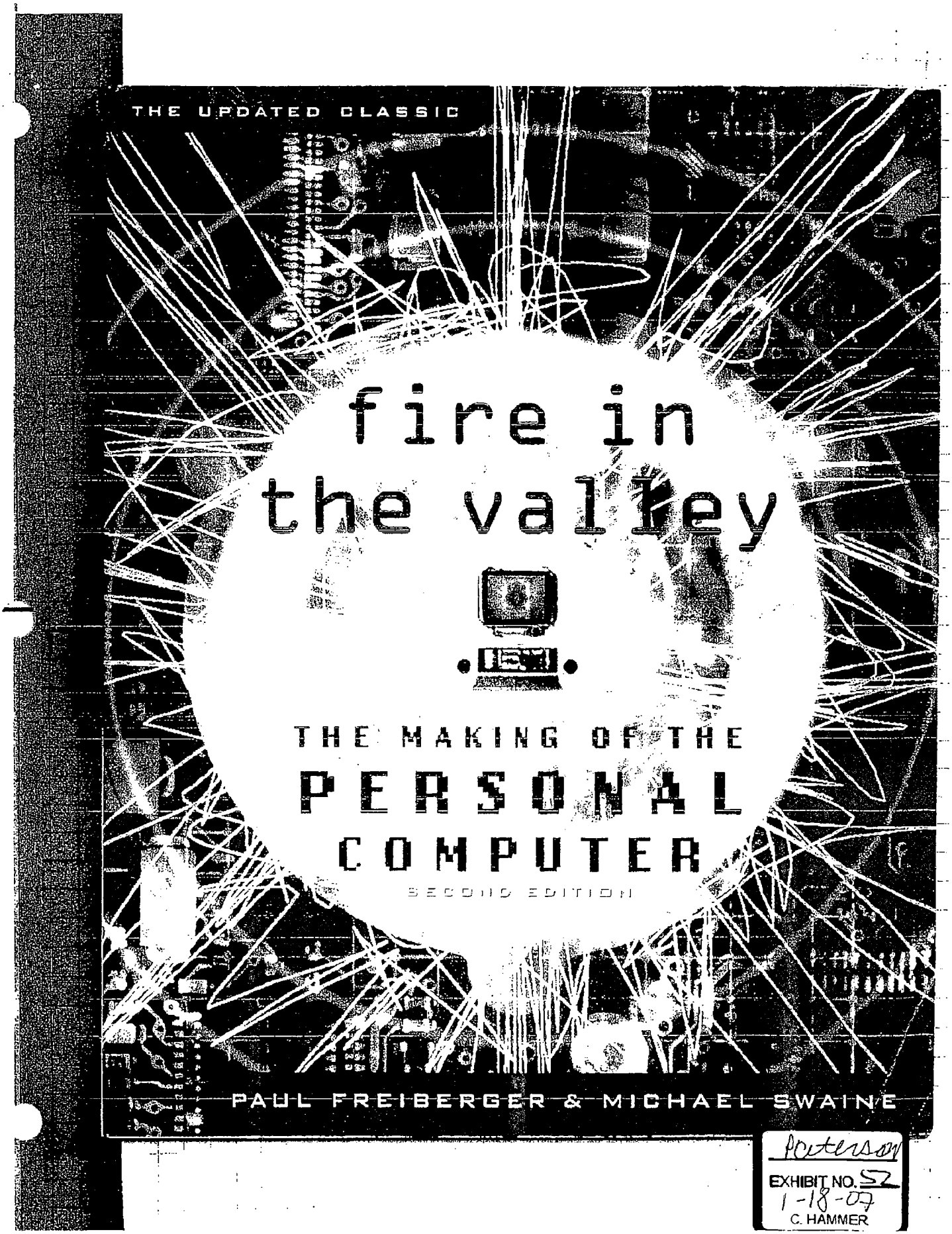


EXHIBIT CC



Peterson
EXHIBIT NO. 52
1-18-07
C. HAMMER

Library of Congress Catalog Card Number: 99-075973

McGraw-Hill

A Division of The McGraw-Hill Companies

1 2 3 4 5 6 7 8 9 0 DOC/DOC 9 0 9 8 7 6 5 4 3 2 1 0 9

ISBN 0-07-135892-7

It was set in Fairfield LH Medium by North Market Street Graphics.

Printed and bound by R. R. Donnelley & Sons Company.

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to the Director of Special Sales, McGraw-Hill, 11 West 19th Street, New York, NY 10011. Or contact your local bookstore.

This book is printed on recycled, acid-free paper containing a minimum of 50% recycled, de-inked fiber.

were sitting in the back of a pickup truck in the Microsoft parking lot discussing the "Apple problem." None of Microsoft's programs would run on the single leading personal computer at that time. Gates shook his head at the thought of converting all his firm's software to work with the 6502 microprocessor that was the brain of Apple's computers. Allen suggested, "Maybe there's a way to do it in hardware."

They brought in Tim Paterson of Seattle Computer Products, located across Lake Washington, to try to build a card for the Apple that would let it run Microsoft's 8080 and Z80 software. They called it the SoftCard. Paterson did a series of prototypes before Don Burdis took over the project. Of course, to run the application software the card also had to run the operating system that the software was written for: CP/M. Gates signed an agreement licensing CP/M for the SoftCard from Digital Research.

One afternoon Allen and Gates sat discussing SoftCard's potential. They agreed that if Burdis could make the SoftCard work, they might sell about 5000 of them. Burdis did make it work, and they sold that many in three months—and many more thereafter.

The SoftCard solved the specific problem of the 6502, but what would happen when the next hot microprocessor came along? Microsoft would have to come up with another SoftCard or translate all its software. In the summer of 1980, Microsoft decided to end its translation nemesis for good. Microsoft approached it by first rewriting all its software into a "neutral" language on a large DEC minicomputer and then writing the chip-specific translator programs that would automatically convert their "neutral" software to the form needed by the 6502 or any other particular processor. The task was massive, but cost-effective if the company intended to supply software to all microcomputer manufacturers and to establish its products as industry standards. That was the idea.

In June, Paul Allen was working on enhancements to a BASIC for machines built around Intel's new 8088 and 8086 chips. The 8086 was one of a fresh generation of microprocessors created explicitly for small computers. It had a more logically designed instruction set and more capabilities for the systems programmer to use. It also possessed a 16-bit architecture; that is, the 8086 could handle information in chunks twice as large as the chunks that the 8080, the Z80, the 6502, or any other common 8-bit microprocessor on the market could handle. This

When it came to the operating system, though, they had an impasse. IBM wanted to buy CP/M outright for $250,000; Kildall was willing to license it to them at the usual $10 per copy rate. IBM left with promises to talk further, but without having signed an agreement for CP/M.

They immediately turned to Microsoft. Gates required no prodding. Once IBM agreed to use a 16-bit processor, Gates realized that CP/M was not critical for their new machine because applications written for CP/M were not designed to take advantage of the power of 16 bits. Kildall had seen the new Intel processors, too, and was planning to enhance CP/M to do just that. But it made just as much sense, Gates told IBM, to use a different operating system instead.

Where that operating system would come from was a good question, until Paul Allen thought of Tim Paterson at Seattle Computer Products. Paterson's company had already developed an operating system, SCP-DOS, for the 8086, and Allen told him that Microsoft wanted it.

At the end of September, Gates, Ballmer, and a colleague took a red-eye flight to deliver the report. They assumed it would determine whether they got the IBM personal computer project. They nervously finished collating, proofreading, and revising the document on the plane. Kay Nishi, a globetrotting Japanese entrepreneur and computer magazine publisher who also worked for Microsoft, had written part of the report in "Nishi English," which, according to Ballmer, "always needs editing." The report proposed that Microsoft convert SCP-DOS to run on IBM's machine. After the sleepless flight, Gates and Ballmer were running on adrenaline and ambition alone. As they drove from the Miami airport to Boca Raton, Gates suddenly panicked. He had forgotten a tie. Already late, they swung their rental car into the parking lot of a department store and waited for it to open. Gates rushed in and bought a tie.

When they finally met with the IBM representatives, they learned that IBM wanted to finish the personal computer project in a hurry—within a year. It had created a team of 12 to avoid the kind of corporate bottlenecks that can drag a project on for years—three and one-half for the Xerox Star, four for the HP-85. IBM president Frank Cary dealt roughly with all internal politics that could cause delays. Throughout the morning, Gates answered dozens of queries from members of

IBM's project team. "They pelted us with questions," said Ballmer. "Bill was on the firing line."

By lunchtime, Gates was fairly confident Microsoft would get the contract. Philip Estridge, who was the project head, an IBM vice president, and an owner of an Apple II, told Gates that when John Opel, IBM's new chairperson, heard that Microsoft might be involved in the effort he said, "Oh, is that Mary Gates's boy's company?" Opel had served with Gates's mother on the board of directors of the United Way. Gates believed that connection helped him get the contract with IBM, which was finally signed in November 1980.

Microsoft first had to set up a workplace for the project, a more difficult task than might be imagined. IBM wasn't just any company. It treasured secrecy and imposed the strictest security requirements. Gates and Ballmer decided on a small room in the middle of their offices in the old National Bank building in downtown Seattle. IBM sent its own file locks, and when Gates had trouble installing them, IBM sent its own locksmith. The room had no windows and no ventilation, and IBM required that the door be kept constantly closed. Sometimes the temperature inside exceeded 100 degrees. IBM conducted several security checks to make sure Microsoft followed orders. Once Microsoft was caught taking a breath, and the IBM operative found the secret room wide open and a chassis from a prototype machine standing outside it. Microsoft wasn't used to dealing with this kind of strictness.

But Microsoft learned. To speed communication between Microsoft and IBM, a sophisticated (for those times) electronic mail system was set up, which sent messages instantly back and forth between a computer in Boca Raton and one in Seattle. Gates also made frequent trips to Boca Raton.

The schedule was grueling. The software had to be completed by March 1981. IBM's project managers showed Gates timetables and more timetables, all of which "basically proved we were three months behind schedule before we started," Gates said.

The first order of business was the operating system. Paterson's SCP-DOS operating system was a close but crude imitation of CP/M. It needed a lot of work to make it fill the bill for the IBM job. Gates brought Paterson in to work on adapting his operating system. The operating system APIs, in particular, had to be completed as soon as possible.

Cloning Around

Had Compaq or IBM changed in '88 or '89, Dell would not have been a factor. Now Dell is driving the industry.

SEYMOUR MERRIN
Computer industry consultant

WHILE APPLE WAS LOSING ITS WAY IN THE WAKE OF IBM'S ENTRY INTO the market, IBM's own fortunes followed a strange path.

When IBM released its Personal Computer, very little about the machine was proprietary. IBM had embraced the Woz Principle of open systems, not at all an IBM-like move. One crucial part of the system was proprietary, though, and that part was, ironically, the invention of Gary Kildall.

Like Michael Shrayer, who had written different versions of his pioneering word processing program for over 80 brands of computers, Kildall had to come up with versions of his CP/M operating system for all the different machines in the market. Unlike Shrayer, however, Kildall found a solution to the problem. With the help of IMSAI's Glen Ewing, he isolated all the machine-specific code that was required for a particular computer in a piece of software that he called the basic input-output system, or BIOS.

Everything else in CP/M was generic, and didn't need to be rewritten when Kildall wanted to put the operating system on a new machine from a new manufacturer. Only the very small BIOS had to be rewritten for each machine, and that was relatively easy.

Tim Paterson realized the value of the BIOS technique and implemented it in SCP-DOS, from which it found its way into PC-DOS.

The BIOS for the IBM PC defined the machine. There was essentially nothing else proprietary in the PC, so IBM guarded this BIOS code and would have sued anyone who copied it.

Not that IBM thought it could prevent others from making money in "its" market. That was a given. In the mainframe market, people spoke of IBM as The Environment, and many companies existed solely to provide equipment that worked with IBM machines. When IBM moved into the personal computer market, many companies found ways to work with the instant standard that the IBM PC represented.

Employees of Tecmar were among the first in the doors of the Chicago Sears Business Center the morning the IBM PC first went on sale. They took their PC back to headquarters and ran it through a battery of tests to determine just how it worked. As a result, they were among the first companies to supply hard disk drives and circuit boards to work with the PC. These businesses took advantage of the opportunity to compete in this market with price, quality, or features. These "parasites" had done something similar with Ed Roberts's Altair six years earlier.

And just as IMSAI had produced an Altair-like machine to compete with MITS, many microcomputer companies came out with "IBM workalikes," computers that used MS-DOS (essentially PC-DOS but licensed from Microsoft) and tried to compete with IBM by offering a different set of capabilities, perhaps along with different marketing or pricing. Without exception, the market resoundingly rejected these IBM workalikes. Consumers might buy a computer that made no pretense of IBM compatibility—Apple certainly hoped so—but they weren't going to put up with any almost-compatible machine. Any computer claiming IBM compatibility would have to run all the software that ran on the IBM PC, support all the PC hardware devices, and accept circuit boards designed for the IBM PC, including boards not yet designed. But IBM's proprietary BIOS made it very hard for other manufacturers to guarantee total compatibility.

Yet the potential reward of creating a 100-percent IBM PC-compatible computer was so great that it was to be assumed that someone would find a way. In the summer of 1981, three Texas Instruments

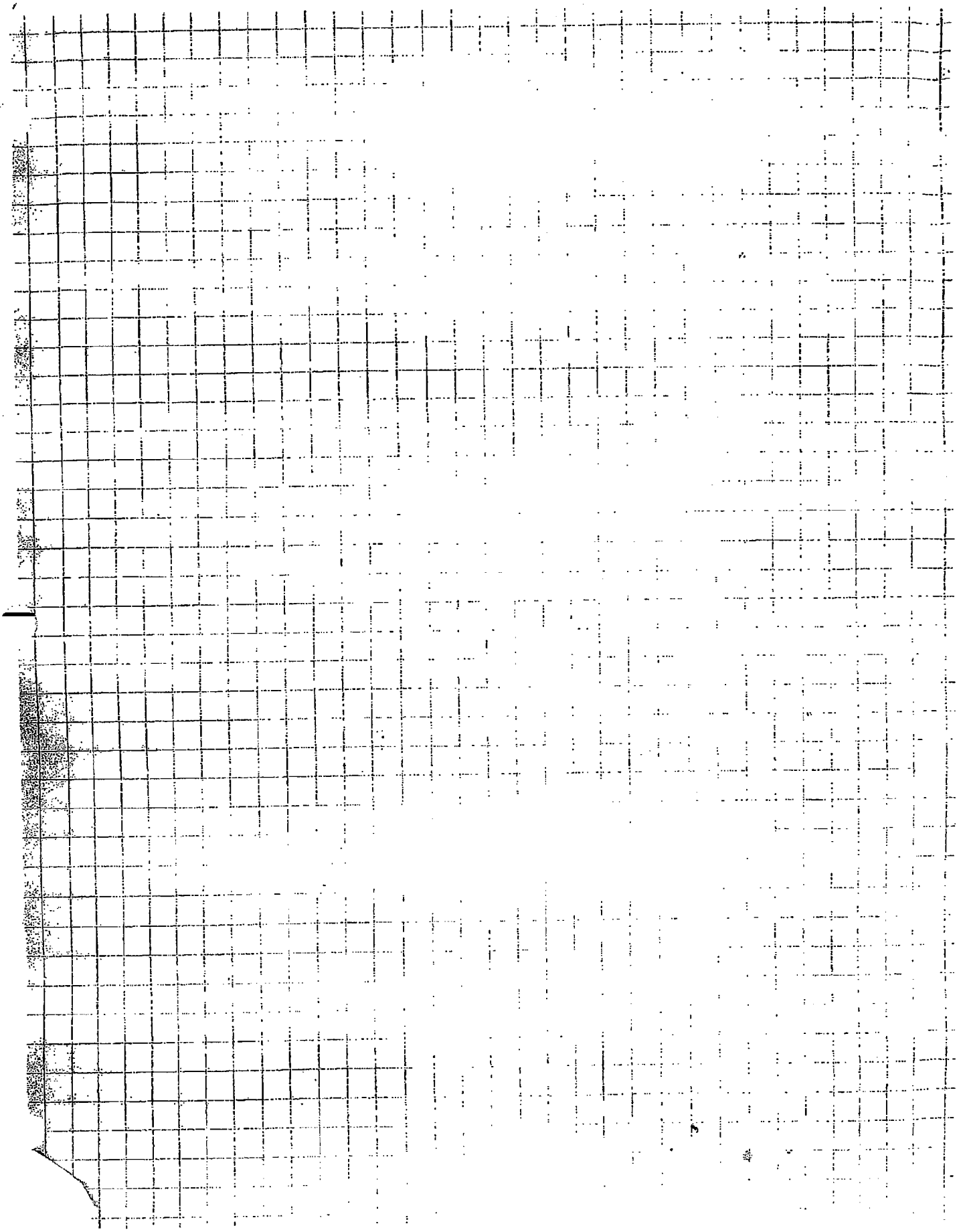


EXHIBIT DD

EFFECTIVE WEB DESIGN: MASTER THE ESSENTIALS

Ann Navarro and Tabinda Khan. 1998. San Francisco, CA: Sybex. [ISBN 0-7821-2278-7. 598 pages, including index. $34.99 (softcover).]

Whenever I pick up a book with a spine that is three-fingers thick, the first thing I ask myself is "Does this book have to be so fat?" It's not that I'm lazy or have something against big books. After all, I've read *Moby-Dick* and *Crime and punishment*. It's just that I think a book should know where it's going, and when it gets there, it should stop. If it can do that, then it can be as fat as it needs to be.

Effective Web design, by Ann Navarro and Tabinda Khan, is a 600-page, three-fingers-thick book. It promises to cover everything the reader needs to know about Web design in one book. It will help readers "understand the practical implications of their design and technical choices to be able to proficiently design for the Web" and to "take the reader step-by-step through the process from concept to creation to making a web site live online" (p. xxi). Does it deliver on that promise? Well, yes and no.

The book is organized into 23 chapters, which the authors call Skills. Each chapter presents a different essential skill that a reader needs to know to effectively design for the Web. Skill 1 discusses the issues that designers face with browsers and compatibility. Skill 23 provides information about techniques for making a site accessible to the visually or physically impaired. In between, there are skills on creating an HTML document, using style sheets, placing pictures, using frames, creating a site navigation plan, understanding search engines, making effective visual presentations, using color on the Web, incorporating audio and video on a site, and attracting and retaining visitors. Each chapter begins by outlining the topics covered and ends by seeing whether you have "mastered the essentials" by reviewing the key points of the chapter.

To its credit, the book covers a comprehensive mix of topics. The writing is accessible, the style draws you in easily, and the explanations are easy to follow. The writers give simple and straightforward descriptions of even the most complex subjects, never talking down to the reader and never taking for granted what the reader may or may not know. Chapter 1 and Chapter 2 provide an excellent introduction to the issues designers face with compatibility because of the different Web browsers in use. Chapter 13, one of the best in the book, offers useful tips on how to plan the design of a Web site. Useful gems of information are sprinkled throughout the book.

But trying to do it all also proves to be the book's greatest weakness. It has trouble sticking to its stated purpose. While Chapters 3 through 12 provide useful instructions in basic Web skills, any good textbook on HTML will provide the same information, and some of them do it better. These chapters never really make clear how having these skills will make you more effective at designing a page. Also, the skills presented in each chapter do not seem to build on each other in any logical or cumulative way.

The book raises some other questions, too. Why does it wait until page 262 to discuss how to plan a Web site's design? In a book called *Effective Web design*, this information should be right up front. Or why does a chapter that discusses the use and effect of color on the Web include only graphics that are black and white? Is this an effective design decision?

There certainly is something for everybody in this book—from the reader who has never surfed a Web site to the reader who is an experienced Web developer—but in the final analysis, the book is fatter than it needs to be. Disturbingly, many chapters lack focus on the stated purpose of the book. So if you're willing to browse through some fat to find the morsels you'll need, this book could be [illegible] However, if you [illegible] that quickly yie[illegible] you'll need to [illegible] pages, I'd look

TWO BOOKS ON SILICON VALLEY

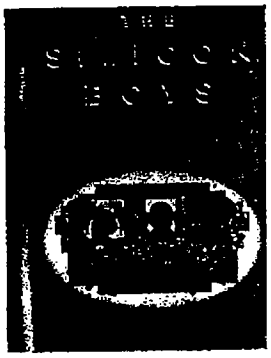
The Silicon boys and their valley of dreams

David A. Kaplan. 1999. New York, NY: William Morrow and Company, Inc. [ISBN 0-688-16148-0. 358 pages, including index. $27.00.]

Nudist on the late shift: And other true tales of Silicon Valley

Po Bronson. 1999. New York, NY: Random House. [ISBN 0-375-50277-7. 248 pages. $25.00.]

Two recent books looking behind the scenes at Silicon Valley take opposite, but equally intriguing, tracks. David Kaplan looks from the top down, detailing the idiosyncrasies of the multi-millionaire and billionaire titans who move markets



and steer companies. Po Bronson concentrates on the lower rungs. He introduces the engineers who move effortlessly from company to company seeking the right challenge. He interviews newcomers who move to Silicon Valley with a suitcase and a dream. Taken together, the two books add significantly to our understanding of the environment.

Estimates claim that 250,000 millionaires live in or around Silicon Valley. Clearly, a million ain't what it used to be, and to rise above the rabble requires more than the usual quirks. Or a billion dollars. *Silicon boys* manages to record quite a few of the oddities that mark the "silicon-aire" crowd. Yachts, castles, and cars are just the recognizable trappings; $18-per-pound ostrich salami is new.

David A. Kaplan is a prize-winning journalist for *Newsweek* who combines a keen sense of the absurd with a journalist's nose for news into one of the best of a large batch of recent books about America's high-tech heroes. Fittingly, Kaplan puts the current silicon rush into perspective by first comparing it with one of the key events of the 19th century—the rush for California's gold in 1849. Just as entrepreneurs such as Levi Strauss, Pullman, and Armour got rich not by mining gold but by mining miners, Kaplan shows how a new crowd of venture capitalists and investment bankers have enriched themselves by mining the Silicon Boys.

For completeness, Kaplan starts with Lee de Forest, the inventor of the "audion," a vacuum-tube amplifier he devised in 1911 not far from Stanford University. His company, Federal Telegraph Company, later begat Magnavox. Then came Frederick Terman, who taught at Stanford and became known as the "Father of Silicon Valley" (p. 33). Two of Terman's students, David Packard and Bill Hewlett, formed Hewlett-Packard on 1 January 1939.

Next in Kaplan's view comes the shockingly incompetent William Shockley. For a man who won the Nobel Prize for his work on the transistor, Shockley was a woeful manager. He was good at hiring, though; he brought in Robert Noyce, Gordon Moore, and six other men who soon left Shockley and founded Fairchild Semiconductor in 1957. There, Noyce invented the integrated circuit. "I was lazy," Noyce said. "It just didn't make sense having people soldering together these individual components" (p. 56). But by 1968 Fairchild was losing momentum, so Noyce and Moore left to form Intel.

Enter Andy Grove. Kaplan reveals that Grove lives sanely compared with other silicon millionaires—he flies coach unless he's saved up enough frequent-flyer miles; he's had the same wife for 40 years; and he's still listed in the telephone book. Still, he's proud of the fact that he's one of the 10 toughest bosses in America. "If Noyce was the spirit of Intel and Moore the heart, Grove was the fist" (p. 64). His annual "Scrooge memo" advised employees to work a full day before the Christmas holiday. He circulated through the sea of cubicles on a Mr. Clean odyssey, and his form of "constructive confrontation" enshrined high-decibel arguing as a management style. Most infamous was the "Late list," which forced employees to sign in if they couldn't drag themselves to work by 8:15 (regardless of how late they had worked the night before).

Kaplan deftly covers terrain that's been written up already. He sets the stage for an in-depth discussion of Steve Jobs and Steve Wozniak, the founders of Apple. Kaplan does some effective reporting, painting pictures that truly come to life. Woz played the part of Merry Prankster, building a box to circumvent AT&T's long distance routers and calling up the Vatican by disguising his voice to sound like Henry Kissinger. Jobs and Woz worked for a spell at HP; then Jobs left for Atari. "The best thing about hiring Jobs," said an Atari executive years later, "is that he brought along Woz to visit a lot" (p. 87). Jobs and Woz ended up creating a single-player version of Pong called Breakout during four all-nighters that gave the young pair mononucleosis.

More good reporting surrounds the legend of Gary Kildall, the father of CP/M. Kildall, who headed Digital Research Inc., wrote the first operating system for the Altair personal computer, and was making an excellent living when IBM came calling one day. Bill Gates had directed IBM to Kildall's Silicon Valley office; he wasn't in the operating system business at the time. But the 38-year old Kildall was enthralled with aircraft, and on that fateful day when IBM knocked, "Gary went flying," as Gates tells the tale (p. 110). An epic piece of Silicon Valley lore, Kildall's friends relate the saga somewhat differently, claiming Gary was with clients (but, yes, he flew to meet them).

At any rate, when IBM went back to Microsoft, the young company saw the opening and went for it. Paul Allen knew Tim Paterson of Seattle Computer Products, who had written Q-DOS (Quick and Dirty Operating System). Q-DOS was basically a rip-off of Kildall's CP/M, but Kildall had never gotten around to suing. For $75,000, Allen bought the rights and renamed it MS-DOS; then he and Gates turned around and leased

it to IBM, retaining ownership and raking in royalties. Kildall grew more and more bitter as Microsoft's fortunes waxed; eventually, he succumbed to alcohol and depression, dying at 52 from a blood clot in his brain resulting from a fall.

Of all the characters in the book, Larry Ellison, founder and CEO of Oracle, comes off the worst. "Ellison is the alpha-male playboy from central casting," Kaplan writes, "tall, thrice-divorced connoisseur of long-legged blondes (preferably employees), defendant in a sexual harassment suit, and the life of any party, as long as he's the center of attention" (p. 119). While Gates' $60-million house in Seattle looks like a Marriott Conference Center, Ellison built himself a 23-acre estate planned by a Zen priest with a Balance-of-the-Elements theme—so refined it uses wooden pegs instead of nails.

In his bunker of an office, Ellison is painted as a caricature of the rich tycoon. One can imagine him as James Bond's arch-villain Dr. No on his island or as a Lex Luthor-wannabe, masterminding the overthrow of Microsoft. Ellison was a master salesman while building Oracle—he may have invented the term *vaporware*. Oracle reps were booking sales for software they not only had never previously delivered but that engineering had never even written. Eventually, that scorched-earth pattern was revealed when the company had millions in bogus sales. Ellison lost $300 million on paper in one fateful day and was no longer a billionaire. He eventually rebuilt the company so that when the Asian monetary crisis hit in late 1997, he lost $2.1 billion in net worth in eight hours but nevertheless remained in the billionaire club. Today Ellison is again the fourth richest man in America, behind Gates, Warren Buffet, and Paul Allen. At 54, he even talks of settling down, although his third wife is against it—"I'm a good ex-wife," she's told friends. "Why does Larry need more of them?" (p. 152).

For all the headline-grabbers that Kaplan introduces us to, there are others with equally interesting stories. He dwells on the labyrinthine financial dealings of the venture capital crowd, detailing the machinations of Kleiner Perkins Caufield & Byers ("KP" for short). Here's where reading *Silicon boys* will serve as a kind of primer for the stories in *The Wall Street journal* and *Business week* that update the latest saga in the Valley. You'll meet John Doerr, whose "home runs" at KP include backing Sun, Compaq, Symantec, Quantum, Cypress, and Lotus. Doerr (or "JD," as he's known) has been described as "the Energizer Bunny on steroids" (p. 187) and has a cell phone built into his ski helmet so that he can work deals while on the slopes.

Another Silicon boy is Jim Clark, founder of Silicon Graphics and Netscape. He's the serial entrepreneur—"if at first you succeed, try, try again." Marc Andreeson, the man who unleashed the World Wide Web when he wrote the Mosaic browser, was recruited to Netscape by Clark, and they both wound up wealthy when Netscape went public. Kaplan takes us behind the scenes as the two men open their third bottle of vintage wine late in the evening while writing the business plan.

What really puts the author's insights into perspective is his own brush with mega-wealth. "In the spring of 1995, I was completing a journalism fellowship at Stanford University. . . . I got to meet, among others, an unknown like Jerry Chih-Yuan Yang" (p. 303). Yang, of course, is the founder of Yahoo! and made a huge pile of money. Yang casually asked Kaplan one day if he was interested in coming to work at the start-up. Well, the young writer would have to relocate his family and go to work for two graduate students in a company with no profits, no revenue, and not even a name on the door. Who knew their market capitalization would reach $44 billion four years later? Kaplan's cut could have been as much as $200 million.

Somehow, that story only makes the book more charming. So does its ability to paint "the largest legal creation of wealth in the history of the planet" (p. 190) in quasi-idealistic terms:

> *"You hear the conventional wisdom around the country that the American Dream is over. I have a problem with that," John Doerr tells any audience who will listen. "Forty percent of GDP growth is from technology. . . . Silicon Valley is a state of mind, to be exported across the nation and around the world. The Digital Universe is just beginning." (p. 191)*

The technology industry is now officially a phenomenon, with countless authors and professors studying and poking for truth in the frenzy. In Silicon Valley, the undisputed capitol of dot-com, the virtual lab is open for inspection.

Po Bronson is the best of a recent crop of talented authors searching for truth in the techno-drama of the Valley. A writer for *Wired* magazine, Bronson records Silicon Valley with a combination of unabashed awe and jaded journalism. He sees the gold rush for what it really is—revolutionary zeal and capitalistic greed grinding together at cyber speed. Yes, these people all want to get rich. They know that the right series of all-nighters can hatch the perfect product, one that catches on like wildfire and burns as bright as a comet. But they also want to change the world, making technology available to bigger and bigger masses, until we all reach a point whereupon truth and justice will naturally follow.

In this way, getting rich can almost be excused, a fact that is important, because it does come first. As Bronson puts it,

The big-picture future—the post-revolutionary future—isn't much on people's minds. Everyone's got a filing due next week or a development milestone or quarter-end sales quota to worry about. There's a great sense that now is the time they will tell their grandchildren about, that today's fever may be the opportunity of a lifetime. (p. 233)

Which is why people continue to stream to Silicon Valley with little more than some cyber savvy and the fear of missing out if they don't make the scene. Bronson likes to frequent the little cafés and parties where the newcomers mix with the old-timers. But he's been around long enough to become slightly jaded, a fact that actually aids his reporting if not his writing. For example, he overhears the party patter behind him and pronounces it "an incomprehensible dialect, a chunky gumbo of snowboarder reggae mumble and high tech's pissing contest '.comming' jargon" (p. 11).

Later, Bronson reveals more of that *Wired* edge while chatting with an earnest entrepreneur-wannabe: "'New business is pop culture, is it not?' he says, soaking in the scene. I'm getting a floating-in-space sensation. This grab bag of cultural references has overloaded my palette" (p. 11).

The danger, it seems, comes when people take themselves too seriously. The book's title comes from a near-mythic legend of a naturist senior programmer who reached an accommodation with his start-up employer that clothing was optional after 10:00 PM. If not for the fact that the programmer was exceptionally gifted, the employer would have declined such an offer, but genius is the true coin of the realm in Silicon Valley, and companies that have a few are in a much better position. Losing one over a silly thing like clothing just doesn't make sense.

The book is organized into a series of chapters concentrating on single themes—"The newcomers," "The IPO," "The entrepreneur," "The programmers," "The salespeople," "The futurist," and "The dropout." This approach keeps the flow choppy and edgy, with Bronson in charge. There's no set pattern; when he covers a salesman, he provides the following glossary:

> Hockey stick disease: *Picture a hockey stick: _/ This is the line graph tracking sales volume chronologically over the quarter—none for two months, then a steep incline at the end. It's not uncommon for 45 percent of the deals to close in the last two days of the quarter.*
>
> "Me too" product: *Software that is pretty much the same as another company's software.*
>
> Dropping your pants: *Lowering your price to close a sale.*
>
> Overhanging the market: *Promising that desired features will be designed into future upgrades.*
>
> Seats: *Seat licenses in volume selling, that is, number of paid users.*
>
> Mindshare: *As opposed to market share. When a salesman says, "I'm building mindshare," what he means is he hasn't sold a thing.*
>
> The Cycle: *The average length of time it takes to make a sale. This can be as long as nine months, which makes it hard when there's a new upgrade and new pricing every six months.*
>
> "The *Queen Mary* has turned around": *Said when a reluctant client finally is ready to buy.*
>
> Spiffs: *The bonus incentives that software firms offer top resellers—mountain bikes, a BMW leased for a year, et cetera. (p. 147)*

Yet nowhere else does Bronson feel the need to compile such a listing. And his instincts are probably correct; this isn't a field manual, after all.

In "The newcomers," we learn about a young man who once grew marijuana to finance a portable keyboard, then ended up taking a programming job and becoming an espresso connoisseur. Next, Bronson introduces us to a programmer friend who has recently landed in the Valley:

> *I ask him if he happens to have a job. He says sure, but he doesn't think he'll last there, it's just a gig, and he's looking for something that takes more brainpower. He started looking the morning of his second day on the job. Sure, the money matters, but day in, day out there's nothing itchier than an unchallenged mind. (p. 9)*

In "The salespeople," Bronson tells the story of the salesman who searches for The Drift-off Moment. This is the point of a sales pitch where the customer loses contact with the pitch.

> *The client's eyes get gooey, and they're staring into space. They're not bored—they're imagining what they could do with SurveyBuilder. All tech salespeople mention this—they've succeeded not when they rivet the client's attention, but when they lose it. (p. 158)*

In "The futurist," Bronson describes a day on deadline with George Gilder, a *Forbes ASAP* columnist, whom Bronson describes as "one of the few technology writers to really do his homework" (p. 167).

> *He's a rogue futurist, contentious and dangerous: he's the Tupac Shakur rap master of futurism . . . he makes hard predictions, predictions that we will be able to look at a few years down the road and ask, Was he right? He starts out tickling your brain, and then, just when your brain starts gig-*

gling, he delivers it a stiff spanking: Pay attention! *(p. 168)*

Bronson imagines the following conversation when George Gilder meets a particularly savvy engineer:

> *GEORGE: Hi, nice to meet you. Hey, that's a sweet access router over there. Wow, both Ethernet and asynchronous ports?*
>
> *STEVE: Yeah, check this baby out —the Ethernet port has AUI, BNC, and RJ-45 connectors.*
>
> *GEORGE: So for packet filtering, you went with TCP, UDP, and ICMP.*
>
> *STEVE: Of course. To support dial-up SLIP and PPP.*
>
> *GEORGE: Set user User_Name ifilter Filter_Name.*
>
> *STEVE: Set filter s1.out 8 permit 192.9.200.2/32 0.0.0.0/0 tcp src eq 20.*
>
> *GEORGE: 1001011011000101110010011101100001010–10100011111001.*
>
> *STEVE: . . . . . .*
>
> *GEORGE: Wait, you lost me there. (p. 173)*

Bronson was an investment banker in a previous life, yet he does not find the need to show off his MBA skills. For example, he admits that he gets lost while listening to an explanation of how newly rich entrepreneurs try to shelter some of their income:

> *Beyond that Allen Damon's explanation gets too complicated for me, having something to do with selling both put and call options and then taking the whole soufflé public. The*

BOOKS RECEIVED 1 JUNE 1999 TO 31 AUGUST 1999

Arnold, Ken, and James Gosling. 1998. *The Java programming language*. 2nd ed. Reading, MA: Addison-Wesley Publishing Company, Inc.

Bunnin, Brad, with Peter Beren. 1998. *The writer's legal companion: The complete handbook for the working writer*. 3rd ed. rev. Reading, MA: Perseus Books.

Druckrey, Timothy. 1999. *Ars electronica: Facing the future*. Cambridge, MA: MIT Press.

Ekman, Richard, and Richard E. Quandt, eds. 1999. *Technology and scholarly communication*. Berkeley, CA: University of California Press.

Gronbaek, Kaj, and Randall H. Trigg. 1999. *From Web to workplace: Designing open hypermedia*. Cambridge, MA: MIT Press.

Head, Alison J. 1999. *Design wise: A guide to evaluating the interface design of information resources*. Medford, NJ: Information Today, Inc.

Hendel, Richard. 1998. *On book design*. New Haven, CT: Yale University Press.

Herman, Jeff. 1999. *Writer's international guide to book editors, publishers, and literary agents*. Rocklin, CA: Prima Publishing.

Hock, Randolph. 1999. *The extreme searcher's guide to Web search engines: A handbook for the serious searcher*. Medford, NJ: Information Today, Inc.

Jaaksi, Ari, Juha-Markus Aalto, Ari Aalto, and Kimmo Vatto. 1999. *Tried & true object development: Industry-proven approaches with UML*. New York, NY: Cambridge University Press.

Jacobson, Robert, ed. 1999. *Information design*. Cambridge, MA: MIT Press.

Joss, Molly W., and Roger C. Parker. 1999. *Looking good in presentations*. 3rd ed. Scottsdale, AZ: Coriolis Group.

Kendall, Kenneth E., ed. 1999. *Emerging information technologies: Improving decisions, cooperation, and infrastructure*. Thousand Oaks, CA: Sage.

Kent, Thomas, ed. 1999. *Post-process theory: Beyond the writing-process paradigm*. Carbondale, IL: Southern Illinois University Press.

Li, Liwu. 1998. *The VisualAge for Smalltalk primer*. New York, NY: Cambridge University Press.

McKay, Everett N. 1998. *Developing user interfaces for Microsoft Windows*. Redmond, WA: Microsoft Press.

Microsoft Corporation. 1999. *Dynamic HTML reference and software development kit*. Redmond, WA: Microsoft Press.

Mohr, Stephen T. 1999. *Designing distributed applications with XML, ASP, IE5, LDAP and MSMQ*. Birmingham, UK: Wrox Press Ltd.

one part I do manage to catch is that TRACEs aren't worth the bother of doing all the paperwork in amounts less than $50 million. Which rules out all but about eighty people in the Valley. (p. 227)

Instead, he marvels at how knowing everyone else's business has become so important. Where he used to fake it and mumble whenever the talk got technical, he now overhears bankers talking about development platforms.

Both Bronson and David Kaplan attended the annual go-kart races that one of the more expensive schools in the Valley uses as a fund-raiser. Where Kaplan saw the "little boy" coming out of all those rich men, Bronson zeroed in more on the drive to be the best at everything. He also put the big donations back into the context of that revolutionary zeal, with an "I'm giving every day at the office" twist:

On the whole, philanthropy seems sort of redundant—they're already giving 70 hour weeks to the creation of new technology meant to empower the world. That's not enough? That said, one's job is still put to the old-fashioned halo test: You've got to be improving society, or what's the point? (p. 225)

But before you can practice hard-core philanthropy, you need the right position, and that often boils down to whom you know. Bronson sees networking as the golden grease that keeps the technology treadmill oiled. Managers can't fill openings if they don't do it; loners won't find work. Bronson was once offered a job after witnessing a company-sponsored soccer match between two teams—the acquirer and the acquisition. Just by knowing what was happening and remarking about it coherently to one of the contestants, Bronson was offered a job on the spot. He declined.

Moniz, Joseph. 1999. *Enterprise application architecture with VB, ASP, and MTS*. Birmingham, UK: Wrox Press Ltd.

Niemeier, Susanne, Charles P. Campbell, and Rene Dirven, eds. 1998. *The cultural context in business communication*. Philadelphia, PA: John Benjamins.

Seshadri, Govind. 1999. *Enterprise Java computing: Applications and architecture*. Cambridge, UK: Cambridge University Press.

Stygall, Gail, and Kathleen Murphy, eds. 1999. *CCCC bibliography of composition and rhetoric*. 1995 ed. Carbondale, IL: Southern Illinois University Press.

Sussman, David. 1999. *ADO 2.1 programmer's reference*. Birmingham, UK: Wrox Press Ltd.

Torrence, Bruce F., and Eva A. Torrence. 1999. *The student's introduction to Mathematica: A handbook for precalculus, calculus, and linear algebra*. Cambridge, UK: Cambridge University Press.

Wempen, Faithe, ed. 1999. *Microsoft Office 2000 Professional fast & easy*. 6-pack ed. Rocklin, CA: Prima Publishing.

Zeff, Robbin, and Brad Aronson. 1999. *Advertising on the Internet*. 2nd ed. New York, NY: John Wiley & Sons.

Accepting the job would have put him too close to the revolution, and Bronson operates better in fly-on-the-wall mode, clandestinely recording for posterity the manic conditions within. What makes him so good is that he keeps it all in perspective, including the effect it has on him. For example, just in witnessing how many revolutions and counterrevolutions are in progress, he starts to crack.

How many revolutions can you join? It's like Monty Python's The Life of Brian: *you can't keep straight the People's Front for Judea from the Judean People's Front. (p. 233)*

Toward the end of the book, in the chapter entitled "Is the revolution over?" Bronson asks,

As the rest of the world adopts the technology being created in the Valley, will the rest of the world also adopt the Valley's work habits and campus parks and organizing principles? Are start-ups and the IPOs and the "total dedication model" not just a way of fostering new technology faster, but a blueprint for redesigning all our industrial paradigm institutions: schools, cities, nation-states? And is it possible, just possible, that if I get any more high-minded than I already am in this paragraph, my brain will explode? (p. 247)

Bronson's Web page at http://www. pobronson.com/ has much more information about him, complete with contact information, discussion groups, and recent writings.

Back in the '60s, Gil Scott-Heron told us, "The revolution will not be televised." It turns out he was right—it will be e-mailed, uploaded, and broadcast via streaming video. One can only hope that authors like Po Bronson stay on the job to report back as the revolution unfolds.

Garret Romaine

EXHIBIT EE

ARTICLE WRITTEN FOR "ENCYCLOPEDIA OF COMPUTERS AND COMPUTER HISTORY" IN 2000.

DOS

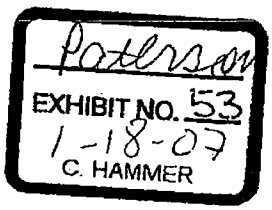
DOS (pronounced Doss) has become the accepted name for the line of operating systems whose names have included QDOS, 86-DOS, IBM Personal Computer DOS, and MS-DOS. At its peak, DOS was by far the most widely used computer program in the world. While at one time "DOS" was a generic term for "Disk Operating System", this is no longer the case, at least within the personal computer industry. ("OS" is now the generic term.)

Although DOS became popular by tagging along with the success of the IBM Personal Computer, its origin actually goes back to an earlier generation of microcomputers. The first widely used microcomputers were built around a chassis called the S-100 Bus. This began with the introduction of the Altair 8800 by Microwave Instrumentation and Telemetry Systems in 1975. The motherboard of the Altair had no active components on it – just a row of 100-pin connectors. The connectors would accept a 5" x 10" circuit board that added a specific function to the computer. The microprocessor itself would be on the CPU card; additional cards would have memory (RAM) and interfaces to a TeleType or keyboard and video display.

In 1978, Seattle Computer Products (SCP) of Tukwila, Washington, was a manufacturer of S-100 memory cards. One of their customers was the only computer store in Seattle at the time, The Retail Computer Store. The store's repair technician, Tim Paterson, was a full-time student at the University of Washington and user of an IMSAI 8080 microcomputer since 1976. When the owner of SCP, Rod Brock, came by the store to make deliveries and take orders, Paterson complained about some problems they were having with the product. After Paterson graduated that June, he went directly to work for SCP to fix those problems. Paterson was the only full-time engineer at SCP, and all design was turned over to him.

In July of 1978, Intel released their new 8086 microprocessor. Brock sent Paterson to an Intel seminar to find out what it was all about. Up until that time, almost all S-100 computers used the Intel 8080 microprocessor or the newer and faster Zilog Z80. Both were 8-bit microprocessors, and could run the same software. The 8086 was a 16-bit microprocessor with the potential to be much faster, although existing 8-bit software would not run on it.

Brock gave Paterson the go-ahead to begin designing an 8086 CPU card for the S-100 Bus, and the first prototypes were working in May, 1979. SCP contacted Microsoft to see about getting 16-bit software for their new computer. As it turned out, Microsoft was fully underway developing software for the 8086, and they were ready to test it on real hardware. Microsoft had moved from Albuquerque, New Mexico, to Bellevue, Washington in early 1979, just a 30-minute drive from SCP's offices. Paterson packed up the prototype and set to work with Bob O'Rear at Microsoft to bring up Stand-Alone Disk BASIC on it.



SCP began shipping their 8086 computer system in November 1979 with Microsoft Stand-Alone Disk BASIC as the only software to run on it. Although BASIC was a suitable programming language for hobbyists to use on their own machines, very little commercial software was written with it. In order to get a software base for their machine that would make it truly useful, SCP needed a general-purpose operating system for it.

Among 8-bit computers, the CP/M operating system from Digital Research had become the standard. Digital Research was known to be working on a 16-bit version for the 8086 microprocessor, CP/M-86, and had expressed interest in using a prototype of the SCP 8086 CPU card to aid in their development (SCP declined). CP/M-86 was expected to be available by the end of 1979.

By April of 1980, CP/M-86 had not yet arrived and SCP was very concerned. Sales of the 8086 computer system were minimal, since only developers or hobbyists who wanted to be on the leading edge would be interested in computer with no real software. Paterson proposed to Brock that SCP take control of the situation by writing their own operating system instead of relying on someone else.

Paterson had graduated with a Bachelor of Science in Computer Science, Magna Cum Laude. Although he had gone directly to work for SCP after graduation, he also dabbled in graduate school. The coursework included a class in operating systems, and he wrote a multi-tasking operating system for the Z80 microprocessor as a term project. He felt qualified to write an operating system for the 8086, and thought he could make it better than CP/M. Paterson proposed a two-phase software development project: first, a quick and dirty operating system, to fill the immediate need for SCP's 8086 computer; second (and never realized), a much more refined operating system available in both a single-user and multi-user version. Again, Brock gave Paterson the go-ahead.

Paterson's primary objective in the design of DOS was to make it as easy as possible for software developers to write applications for it. To achieve this, Paterson sought to make the Application Program Interface (API) compatible with CP/M. While a given 8-bit program written for CP/M could not be directly run on the 16-bit 8086, it was possible for that program's author to translate it in a semi-automated process so that it would. CP/M compatibility of the API was key to making the translated program run correctly. Also, it was hoped that the familiarity of the CP/M-style API would make it easier for developers to learn to write programs for DOS.

The secondary objective in the design of DOS was to make it fast and efficient, so it was written entirely in 8086 assembly language. Paterson was particularly concerned about the way files were organized on disk; he felt that the format used by CP/M was a significant bottleneck. After evaluating techniques used by Unix, ISIS, UCSD P-System, and others, he settled on a variation of the system used by Microsoft Stand-Alone Disk BASIC. It used a File Allocation Table (FAT), which was extremely compact. To Paterson, it seemed quite suitable for the 1.2 MB floppy disks of the day, and could handle disks up to 64 MB, if microcomputers ever needed anything of that size!

Over the months from April through July, 1980, Paterson was able to spend about half his time working on QDOS, the Quick and Dirty Operating System. It began shipping with the 8086 computer system in August. SCP approached Microsoft about adapting their software to run under DOS, who said it was possible – for a price.

Shortly afterward, Microsoft came back to SCP with a different proposal. Microsoft offered to market DOS for SCP, and they already had the first customer lined up, although they couldn't reveal who it was. They made a deal: Microsoft would pay SCP $10,000 for the right to market DOS, and $15,000 for each OEM customer. The per-customer figure was half of what SCP figured was the going rate for a flat-fee license, which was a common arrangement at the time. So SCP came away with $25,000 in cash, and Microsoft had obtained an operating system for their secret customer, IBM.

Microsoft had been with working with IBM on their personal computer project since the outset. Microsoft originally sent IBM to Digital Research for the operating system, but IBM felt rebuffed when Digital Research would not sign a non-disclosure agreement. So Microsoft offered them an alternative by striking the deal with SCP for DOS.

In July of 1981, the month before the IBM Personal Computer was announced, Microsoft offered to buy DOS (now called 86-DOS) from SCP instead of continuing to pay a $15,000 per-customer royalty. This would give Microsoft flexibility in pricing, and return SCP back to its roots as a hardware company. Microsoft paid $50,000, plus a license for SCP to include DOS with their computer systems. Five years later, Microsoft and a struggling SCP fought a legal battle over the specifics of that DOS license; in the end, it was settled by Microsoft buying the license back for a reported $975,000. Thus Microsoft's payments to SCP for DOS ended up totaling $1,050,000.

In the early days of the IBM PC, DOS was viewed as IBM's proprietary operating system. Microsoft set out to change that in 1982 by trying to interest their OEM customers – who were primarily buying Microsoft's programming languages – to sign up for DOS as well (now called MS-DOS), in direct competition with CP/M-86. Although Microsoft had previously committed to developing CP/M-86 versions of their products, they were eventually able to convert every customer to DOS versions instead – in one case, by simply giving DOS away. That saved them considerable development effort, and at the same time made CP/M-86 less attractive since it didn't run any of Microsoft's software.

Microsoft's marketing combined with the success of the IBM PC and compatibles made DOS a runaway hit for 15 years. Microsoft kept improving and evolving it, often by including in DOS features that had been available in programs from third parties. The beginning of the end came in 1995 with Microsoft's release of Windows 95, which had the function of DOS built in. Microsoft stopped updating DOS after that, as part of a strategy to move from the 16-bit DOS world to a new 32-bit world of Windows 95 and Windows NT.

Bibliography

Manes, Stephen, and Paul Andrews. *Gates: How Microsoft's Mogul Reinvented an Industry - and Made Himself the Richest Man in America*. Doubleday, 1993.

Wallace, James, and Jim Erickson. *Hard Drive: Bill Gates and the Making of the Microsoft Empire*. John Wiley & Sons, Inc., 1992.

Duncan, Ray, editor. *The MS-DOS Encyclopedia*. Microsoft Press, 1988.

Paterson, Tim. "An Inside Look at MS-DOS." *Byte*, Vol. 8 No. 6 (June 1983), pp. 230 - 252.

Hunter, David. "Tim Paterson: The Roots of DOS." *Softalk for the IBM Personal Computer*, Vol. 1 No. 10 (March 1983), pp. 12 - 15.

EXHIBIT FF

THE TIMES HIGHER FEBRUARY 8 2002

Tony Durham

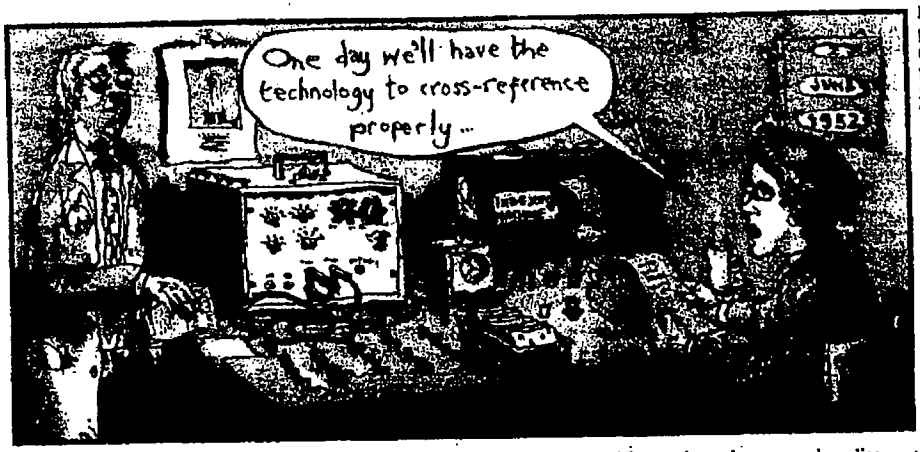
Iffy index no help to fuzzy logic

Since 1945, computers have permeated our culture, economy and physical environment. For a book with the words encyclopedia, computer and history in its title, the problem is not what to put in but what to leave out.

Raúl Rojas explains in his preface that this is only an introduction to "the fascinating world of computer history". It focuses on the history of systems and networks, with a limited amount of material on software and the theory of computation. Some material on contemporary digital media and cyberculture has been included, as if to show where, astonishingly, it all led.

By claiming to cover the entire history of computing, the encyclopedia is perhaps easier to sell than one that ends at, say, 1990. But the rather patchy recent material adds little to its use as a reference source.

There are some good factual articles, such as Chris Woodford's pieces on *Wired* magazine and Netscape Communications. There are snippets on ASCII art, chat rooms, virtual communities, spam, Yahoo!, netiquette and emoticons. The smiley face :-) was apparently first used by Scott Fahlman, a researcher at the Massachusetts Institute of Technology. Now there's a guy who has made the world a happier place. But the book often lacks the historical dimension: colour management is discussed without reference to Edwin Land's revolutionary work on colour perception, and the article on fonts fails to mention that typographers were creating founts (spelt thus) long before computers appeared.



Encyclopedia of Computers and Computer History

Edited by Raúl Rojas

Fitzroy Dearborn
930pp, £190.00
ISBN 1 57958 235 4

Much more secure is the material on early hardware. Leibniz and Pascal both made calculating machines in the latter part of the 17th century. Less well known is Gottfried Schickard, a friend of the astronomer Kepler, who invented a calculating machine in 1623. Schickard's letters were lost for 300 years.

Vigorous debate breaks out when we reach the 20th-century pioneers. For each of the major figures, Rojas has been able to find a passionate but scholarly advocate, for example Andrew Hodges on Alan Turing. Rojas himself speaks up for Konrad Zuse, who began building computing machines in Berlin in 1936. But the computer is not a single invention. Even such key ideas as the stored program evolved gradually from theoretical abstraction to engineering reality.

At the back of volume two, there is a list of all known computers as of 1955. At that time, 16 countries had built computers. From 1955 onwards, the book concentrates on machines that are seen as commercially or technically important. For a while, the mainframe business was dominated by a group of companies known as "IBM and the seven dwarfs". Then Digital Equipment led the minicomputer revolution, and Apple made the breakthrough in personal computers.

There are articles on the major personal application categories such as spreadsheets, word processing and desktop publishing.

Ivan Sutherland's invention of the virtual reality headset is mentioned not in his biography but in the general article on VR. The article on the "Reduced instruction set computer" does not mention an important Risc feature, the elimination of microcode, but in the article on "Microprogramming", however, all is made plain. This article contains a better explanation of the whole Risc concept.

The article on searching and sorting mentions only one sorting algorithm, the insertion sort. The efficient and widely used Quicksort algorithm was invented by Tony Hoare and is duly mentioned in his biography, but with only the barest explanation. You might not guess that elsewhere in the book there is a full, implementable description of Quicksort. It is given as an example in the article on algorithms.

"Cray I", "Cray Research" and "Cray, Seymour" come one after another, by three different authors and with much repetition. The article on supercomputers goes over the Cray saga yet again, when it could have been telling us more about Japanese supercomputers and explaining how vector processing and pipelining work.

There are large overlaps between "Wiener,

Minicomputers were distinguished by their physical size (page 489 — wrong) and by their word length (page 551 — correct). A twisted-pair cable has four wires on page 537 and, more plausibly, two wires on page 781. The 6502 microprocessor was made by Motorola (page 849); by Rockwell (page 520); by MOS Technology (page 409). In fact, the 6502 was designed by MOS Technology, though a number of other firms produced it under licence.

The essay-like articles are richly cross-referenced and fine for surfing. But if you want the answer to a specific question, you are compelled to use the index. And here the book does itself a real injustice because, unless my sampling was very unlucky, the index is distressingly incomplete.

Jef Raskin is mentioned in the Macintosh entry but not in the index. Clive Sinclair is absent from the index, even though Jonathan Bowen very properly mentions the Sinclair ZX80 in his article on microprocessors. The ZX80 showed for the first time how small and cheap a microcomputer can be.

The cult programming language Forth is not indexed. But Woodford gives it its place in history as the forebear of PostScript, the language in which computers talk to printers. The Objective-C language is rightly mentioned as an important part of the NeXT computer system. Not a peep about it in the index, though. Neither Symbolics nor Tandem is in the index. You would have to chance on the articles on Lisp and fault-tolerant computing respectively to discover what the two companies did.

Or suppose you wanted to find out about mercury delay-line memory. There is nothing about it in the index under "mercury", "delay" or "memory". But the book contains at least three mentions and one picture of delay-line memory. Turing at the National Physical Laboratory wanted some for the Ace but could not get them delivered in time. Maurice Wilkes was luckier, or more patient, and his Cambridge team succeeded in building the Electronic Delay Storage Automatic Calculator. Martin Campbell-Kelly's article on Edsac has the details, and there is more in Jon Agar's biography of Wilkes. The bulky piece of plumbing that we see Wilkes admiring is in fact a group of delay lines filled with the toxic liquid metal

There are some good factual articles, such as Chris Woodford's pieces on *Wired* magazine and Netscape Communications. There are snippets on ASCII art, chat rooms, virtual communities, spam, Yahoo!, netiquette and emoticons. The smiley face :-) was apparently first used by Scott Fahlman, a researcher at the Massachusetts Institute of Technology. Now there's a guy who has made the world a happier place. But the book often lacks the historical dimension: colour management is discussed without reference to Edwin Land's revolutionary work on colour perception, and the article on fonts fails to mention that typographers were creating founts (spelt thus) long before computers appeared.

Encyclopedia of Computers and Computer History

Edited by Raúl Rojas

Fitzroy Dearborn
930pp, £190.00
ISBN 1 57958 235 4

Much more secure is the material on early hardware. Leibniz and Pascal both made calculating machines in the latter part of the 17th century. Less well known is Gottfried Schickard, a friend of the astronomer Kepler, who invented a calculating machine in 1623. Schickard's letters were lost for 300 years.

Vigorous debate breaks out when we reach the 20th-century pioneers. For each of the major figures, Rojas has been able to find a passionate but scholarly advocate, for example Andrew Hodges on Alan Turing. Rojas himself speaks up for Konrad Zuse, who began building computing machines in Berlin in 1936. But the computer is not a single invention. Even such key ideas as the stored program evolved gradually from theoretical abstraction to engineering reality.

At the back of volume two, there is a list of all known computers as of 1955. At that time, 16 countries had built computers. From 1955 onwards, the book concentrates on machines that are seen as commercially or technically important. For a while, the mainframe business was dominated by a group of companies known as "IBM and the seven dwarfs". Then Digital Equipment led the minicomputer revolution, and Apple made the breakthrough in personal computing. Do not look here for minor brands of PC clone or Unix box. Even ICL, the British company that nipped at IBM's heels, is barely mentioned.

The book does not set out to be an encyclopedia of software, but programs and programming are an integral part of computer history. There are substantial articles on historically significant operating systems such as Multics, Unix, Linux, VMS, Windows and Windows NT, CP/M and DOS. To get Brian Kernighan to write on Unix is a coup.

A general article on operating systems, by John Deane, tells how Unix inspired CP/M, "which was copied as QDOS, then reworked by Microsoft for IBM's personal computer". The "DOS platform" article tells the tale in more detail. QDOS was no mere copy. It had a CP/M-like interface, but handled files more efficiently. It was written for Seattle Computer Products by a student at the University of Washington, Tim Paterson. A small local company called Microsoft offered to market QDOS, claiming that it already had one prospective customer. "SCP came away with $25,000 in cash, and Microsoft had obtained an operating system for their secret customer, IBM." Of course, Microsoft never looked back. The story is told here by Paterson himself, without obvious bitterness.

Turning to programming languages, the coverage is adequate but hardly encyclopedic. There is material on Ada but not Coral; APL but not Forth; Oberon but not Objective-C. Individual application packages generally do not get their own entries, but there are articles on the major personal application categories such as spreadsheets, word processing and desktop publishing.

Ivan Sutherland's invention of the virtual reality headset is mentioned not in his biography but in the general article on VR. The article on the "Reduced instruction set computer" does not mention an important Risc feature, the elimination of microcode, but in the article on "Microprogramming", however, all is made plain. This article contains a better explanation of the whole Risc concept.

The article on searching and sorting mentions only one sorting algorithm, the insertion sort. The efficient and widely used Quicksort algorithm was invented by Tony Hoare and is duly mentioned in his biography, but with only the barest explanation. You might not guess that elsewhere in the book there is a full, implementable description of Quicksort. It is given as an example in the article on algorithms.

"Cray I", "Cray Research" and "Cray, Seymour" come one after another, by three different authors and with much repetition. The article on supercomputers goes over the Cray saga yet again, when it could have been telling us more about Japanese supercomputers and explaining how vector processing and pipelining work.

There are large overlaps between "Wiener, Norbert", "Cybernetics" and a third article on Wiener's writings. Peter Asaro's clear cybernetics article covers virtually all leading cyberneticists and makes the necessary link to present-day genetic algorithms and artificial neural networks. Asaro argues that the main ideas of cybernetics were presented by Ross Ashby in 1940, three years before Wiener. It was Wiener who eventually gave the field its name, but devoting two more articles to Wiener seems excessive.

The editors have let through a few inaccuracies and some shaky technical explanations. The book sometimes contradicts itself. than Bowen very properly mentions the Sinclair ZX80 in his article on microprocessors. The ZX80 showed for the first time how small and cheap a microcomputer can be.

The cult programming language Forth is not indexed. But Woodford gives it its place in history as the forebear of PostScript, the language in which computers talk to printers. The Objective-C language is rightly mentioned as an important part of the NeXT computer system. Not a peep about it in the index, though. Neither Symbolics nor Tandem is in the index. You would have to chance on the articles on Lisp and fault-tolerant computing respectively to discover what the two companies did.

Or suppose you wanted to find out about mercury delay-line memory. There is nothing about it in the index under "mercury", "delay" or "memory". But the book contains at least three mentions and one picture of delay-line memory. Turing at the National Physical Laboratory wanted some for the Ace but could not get them delivered in time. Maurice Wilkes was luckier, or more patient, and his Cambridge team succeeded in building the Electronic Delay Storage Automatic Calculator. Martin Campbell-Kelly's article on Edsac has the details, and there is more in Jon Agar's biography of Wilkes. The bulky piece of plumbing that we see Wilkes admiring is in fact a group of delay lines filled with the toxic liquid metal. A memory leak in those days was not a matter for debugging but for decontamination.

Altogether this encyclopedia is a bit of an Easter egg hunt. It has some excellent material, but a book of this kind must be tighter, more comprehensive, more carefully checked and more brutally edited. Above all, it needs to be better indexed. All these things could be achieved in a second edition. Which reminds me: though recursion is explained, there is no entry for "iteration".

Tony Durham is on the staff of *The THES*.

First Impressions

This week's competition, in which you have to identify a book from its opening sentence, is from a macabre novel of teenage fantasy:

"I had been making the rounds of the Sacrifice Poles the day we heard my brother had escaped."

■ Entries, including postal address, should be sent to First Impressions, *The THES*, Admiral House, 66-68 East Smithfield, London E1W 1BX, faxed to 020 7782 3300 or emailed to theschat@thes.co.uk

The winner receives a £25 Blackwell's voucher and the closing date is February 12.

The winner of last week's competition, who correctly identified Lord Beaverbrook's *The Abdication of King Edward VIII*, is Christopher Wood of London.

professional c
scientist's tool
Based on a
and more exp
single-volume
ence book on
tionary compr
these two volt
comprise a ta
authoritative
for programm
ers. Volume o
required back

Buildi

Pro