*The* **Analytical Engine**

JOURNAL OF THE COMPUTER HISTORY ASSOCIATION OF CALIFORNIA

Volume 3.4 **Cuthbert Hurd, 1911-1996** Fall 1996

Fall 1996 Volume 3.4

The Analytical Engine

JOURNAL OF THE COMPUTER HISTORY ASSOCIATION OF CALIFORNIA

## Editorial: NOT EASY, BUT FUN

The CHAC has turned a rewarding full circle. When ANALYTICAL ENGINE 1.1 appeared, in the summer of 1993, we wrote editorials full of questions. What was the future of computer history in the Bay Area? How much support could we count on for a local computer museum? How much informal coordination was there among private collectors? Would universities cooperate? Would ex-employees rally to save artifacts and intellectual property from ghost companies? Who, in a word, cared?

We finally have answers—all very positive. In the last issue of the ENGINE, Fred Davis set forth the outline of the San Francisco Computer Museum as planned by the Computer Institute; in this issue, Dr. Len Shustek gives us a progress report on the exciting new Computer Museum History Center recently established in Santa Clara. In short, the amount of energy devoted to the history of computing in Northern California is at last roughly in proportion to the crying need.

Local computer industry workers are realizing that they, too, are part of history and can do honor to the history they made. You'll find an excellent example on page 26 where Don Thomas, in his article "Did You Hear Anyone Say Goodbye?", gives a firmly proactive assessment of the sad fate of Atari, the great ST, and the Jaguar game deck. Also enjoy this issue's interview with John Sell, who graces the ENGINE's pages with the straight —and little-known—story of workstation maker Ridge Computers.

Yet the CHAC and its colleagues still have a lot to accomplish. What a relief and privilege, then, to announce at last that the CHAC has an actual staff—all volunteers, all dedicated, and all quite indispensable. These are the people who have kept your (much busier!) Association running during the recent volatile months when Bay Area computer history has been bootstrapping itself.

*Edwin Vivian El-Kareh*, Tactical Director, is the Engineering, Marketing and Sales Director for AB Networks in Sunnyvale. He has taken primary responsibility for the CHAC's storage since the arrival of the SDS 930, and improved its conditions materially by repacking and rearranging artifacts, putting up shelving, substituting durable plastic crates for deteriorating cardboard boxes, leading work parties and inventories, and generally keeping our collection near the top of the stack. Edwin is a master of what might be called the physics of function, and can give incisive opinions on everything from cranes and riggers, to solar electricity, to industrial adhesives....name it and he's done it! In the spare time he always seems to have (but how?) he also serves as Secretary of the Perham Foundation and Vice-President of the Silicon Valley Engineering Council.

*Erich W. Schienke*, Assistant Editor of the ANALYTICAL ENGINE, is an alumnus of the Hampshire College Program in Science, Technology and Society. He recently arrived in California craving "hands-on experience in computer history" and is certainly getting it, not only as the ENGINE's newest staff member but as General Manager of the Computer Institute. Erich has also been working at the noted George Coates PerformanceWorks theater, composing and playing electronic music, and staying involved with cutting-edge projects in San Francisco's Multimedia Gulch. Adept with Adobe Photoshop, Erich was kind enough to tune this issue's cover photo of the late Cuthbert Hurd (see p. 21), and he also contributed the obituary of Seymour Cray and the review of *Where Wizards Stay Up Late*, Katie Hafner and Matthew Lyon's widely enjoyed history of the Internet. Look for more of Erich's work in forthcoming issues, including a feature article in 4.1.

*Hilary L. Crosby* CPA, Volunteer Coordinator, is a partner in the San Francisco accounting firm of Crosby & Kaneda, serving nonprofit organizations exclusively. She has been involved with Bay Area nonprofit agencies, as both an administrator and a consultant, since 1978; in 1990 she received her master's degree in nonprofit administration (MNA) from the University of San Francisco.

Hilary's assistance has been invaluable to the CHAC since 1993, when she helped us prepare and file the inch-thick paperwork required for our nonprofit status. As Volunteer Coordinator she performs recruitment, follow-up and assignment for numerous volunteer activities, including work parties and inventory of our artifacts.

*Mairi Ross,* Public Relations Director, distributes the Association's press materials and maintains press and Internet contact. She is an Account Executive for Web research and audience development at Cybernautics, a leading web design and online promotion company in Sausalito. She has nine years' experience in public relations, marketing, and copywriting; her primary areas of focus on the Internet are women and technology, and global marketing. Before joining Cybernautics, she was the president of her own award-winning PR firm with clients that included Saturn Cars, Bedford Properties, and the government of Greece. She has lived in France, Mexico and India before arriving in California. She is a member of Women in Multimedia, The Computer Institute, and Webgrrls.

## YES, ADS

Beginning with issue 4.1, the ENGINE will accept advertising, for which the rationale is quite simple. If this magazine keeps publishing one interview per issue, tries to include at least one full-length feature, keeps up with the latest computer history news, continues to allocate space for queries, and throws in a few pictures, we'll be publishing issues with a minimum of fifty to sixty pages. What we fondly call a "fat" or "double" issue is really as big as *every* ENGINE could be. But—as we've often complained—we don't raise quite enough money to publish at that length time after time.

We want to avoid raising the price, which most people seem to find fair as it is. Newsstand sales involve such a thin margin that if even one customer stiffs us—which has happened—the profit from bulk sale of an entire issue can be wiped out. Our mailing list continues to grow, slowly, but *we intend to publish a substantial magazine no matter how many subscribers we have.* Therefore, the ENGINE needs additional income from a separate source. It won't take much; if each issue includes one page of advertising for every six to eight pages of content, the magazine will be well buffered against variations in circulation. The ENGINE is one of the CHAC's most important accomplishments and deserves to be self-sustaining.

We look forward to publishing ads for hardware and software companies, vendors and auctioneers of scientific instruments, organizations and publications concerned with technical history, and other entities interested in reaching the ENGINE's readers specifically. Readers who wish to buy or sell artifacts will be pleased to know that we're (finally!) offering classified ads which will be kept as inexpensive as possible. The ads we accept will be appropriate, in content and tone, to the ENGINE's focus and overall direction.

If you'd like to take out an ad in the ENGINE, e-mail your fax number or snail-mail address to engine@chac.org and we'll send you a rate card and contract. Discounts are available for multiple insertions; at the moment we're quoting rates only for b&w, but if the response warrants it, we'll set rates for color—and then, of course, we could use color elsewhere in the magazine....

## TZL EOF (sigh)

David McGlone has announced, "with a heavy heart and great reluctance," that *The Z-Letter,* one of the premier resources for admirers of CP/M and ZCPR3, has ceased publication with issue #41. "Renewals and new subscriptions are down to almost nothing," he writes, "and orders for software, books and manuals are no longer coming in fast enough to make up the difference." Several other factors enter into his decision, including the prevalence of new "distributions" of CP/M for which no royalty is paid to Novell.

Material from the Lambda archives will remain available, since David plans to produce CD-ROMs of legal and public-domain software, manuals, third-party literature, schematics, and digitized photos of hardware. For details, e-mail David at mcglone@efn.org.

Personally, we found *The Z-Letter* one of the most honest and substantial of the user newsletters we received, and brimming with detail especially at the bare-metal level. We were in the habit of reading each issue as soon as it arrived, and will miss it.

## SOURCE TO THE PEOPLE!

Long-time correspondent Doug Jones reports that, thanks to the efforts of Bob Supnik at DEC, source for three operating systems—DEC's OS/8 for the PDP-8, SCO UNIX for the PDP-11, and Data General's OS for the DG Nova—is now freely available subject to non-commercial public license downloadable with the code. The OS/8 license is partly hypothetical since some parts of the OS are not known to exist; if you have OS/8 source tapes in your collection we urge you to contact Doug at jones@pyrite.cs.uiowa.edu.

Caldera Inc., a developer of Linux technologies, acquired several PC DOS operating systems from Novell, Inc. on July 23, including CP/M®, DR DOS®, PalmDOS®, Multi-User DOS® and Novell DOS 7®. Pending an evaluation and organization of the technologies, source code for all of these will be made available from Caldera's web site during the first quarter of 1997. Caldera OpenDOS, an enhancement of Novell DOS 7, is available now at http://www.caldera.com/dos/download.html; non-commercial use is free.

## VIRTUAL VARIETY....

The World Wide Web hasn't changed everything for us—we still pick two newspapers off the door-mat every morning, and shop at the supermarket—but it sure has changed most things to do with computers and the way we use them; and it's about to change the CHAC in a big way.

The existing CHAC web page, frankly, is dusty deck; it hasn't been materially revised or expanded in months. Yet it's received about 8,500 hits in the last year and its average hit rate is increasing steadily. Jim Clark was right years ago, the Web is a major publishing medium bringing about global changes in delivery of information—at least.

An evolution thus begins with the seminal quote from Erwin Tomash: "Of all the ways of teaching history, a museum is the most expensive." If we take that as well demonstrated by evidence, what does it mean about the future course of the CHAC?

Certainly the CHAC has a physical collection. Just as certainly, it doesn't need to build a physical museum. The Computer Museum History Center and the San Francisco Computer Museum are two very different institutions with (proposed) sites in separate regions, but if everything now planned comes to fruition, they should complement each other to make the Bay Area a world center and destination for computer history. Okay. Kudos. Deep sigh. Our painful concern of three and four years ago, that Northern California might never have a truly professional and comprehensive computer museum, is laid to rest.

That being so, the CHAC can establish a historical and educational resource that *isn't* expensive—at least not on the same scale—and *doesn't* correlate closely to the location of its artifacts, or not in the same sense. Through the ENGINE we will keep you closely informed of this resource's evolution. At the moment we're tossing around ideas like a database for repairers and restorers, and exploring involvement in K-12 education, which currently includes little if any material on the history of computing. While we're tossing we mean to develop an expanded Web site celebrating the history of computing in California and showcasing, but hardly limited to, the CHAC's own collection.

We've got all the bits. First and foremost, we've got talent; the current CHAC staff boasts three experienced HTML coders and, naturally, volunteers beyond those. We've accumulated equipment for audio and video recording and (thanks to the cooperation of perennial CHAC supporter Tom Ellis) recently added a Web-quality digital still camera. Our recent acquisition and testing of Corel Ventura 7—at long last, an MS-Windows version that doesn't crash every ten minutes—may answer our plaintive wishes for serious layout software with output to electronic document formats. Finally, no praise is too great for our Internet service provider, our neighbors WombatNet, whose combination of reliability, connection quality, support quality and fair pricing has made a qualitative contribution to the CHAC's presence in cyberspace.

Your Association, therefore, will spend 1997 expanding its publishing operations and positioning itself as a Web content provider. We must also stabilize the storage of our artifacts, and we hope to increase the number of ENGINE subscribers substantially. If we can accomplish all that—not easy, but fun—the CHAC will enter its fifth year as a truly robust force for the guardianship of computer history.

## CALCULATOR CATALOG!

Guy Ball and Bruce Flamm, the tireless directors of the International Association of Calculator Collectors, have produced *The Complete Collector's Guide to Pocket Calculators: Prices, History, Models.* Going far beyond the simple "price guide" format, the *Collector's Guide* presents a history of portable calculators from the early 1960s to 1979, calculator production dates, dimensions, distinguishing characteristics and countries of origin. One section is devoted to the "Cal-Tech," built by Jack Kilby, Jerry Merryman and James van Tassel in 1967 as a proof of concept for Texas Instruments; weighing almost three pounds, this was *not* destined for anybody's pocket, but it was and is the earliest IC-based calculator, with four functions, memory, and ticker-tape thermal printout like the later Canon Pocketronic. This is only one example of the historical detail that abounds throughout the book. A selection of vintage calculator ads rounds out the included material.

The catalog section lists over 1,500 models from over 220 manufacturers in a surprising number of countries. Almost 500 calculators are photographed. This is an ideal companion to Haddock's *Collector's Guide* and, like the earlier book, will quickly become indispensable to those working (or playing) in the field.

Price of this 204-page quality paperback is US$23.95. Order directly from the publisher:

Wilson/Barnett Publishing
Box 345
Tustin CA 92781-0345 USA.

## THE IMPORTANCE OF SOFTWARE HISTORY

**by Luanne Johnson**

The history of software is significantly underrepresented in the history of computing as it is currently being addressed. Popular books, academic works and other sources for the history of computing tend to focus on the developers of hardware and on the entrepreneurs who turned laboratory discoveries into commercial realities. By contrast, popular books about the development and developers of software—if we exclude Bill Gates and Microsoft—are few, academic research is sparse, and the development of archives (and archiving protocols) has become a concern only recently.

This leads to the present distorted reflection of historical truth which underrates the contribution of software to today's incredible market demand for computing power. Advances in hardware technology have been paralleled by the explosion of innovative applications for each new platform—applications which, for the most part, arise from continual software development and entrepreneurial marketing. Software in the United States has become a multibillion-dollar industry, the envy of the developed world, and it is odd that so little effort in proportion has been spent on organizing and preserving the pertinent history. We urgently need, for example, an exhibit tracing software development and demonstrating the ongoing relationship between new hardware platforms and new capabilities in software. I think it would be useful for aspiring entrepreneurs to understand the value of perseverance in development—to realize that most software pioneers *weren't* instant successes, that they often had not much more than good ideas backed up by scanty capital obtained with great sacrifice. Yet perseverance rewarded them with companies worth hundreds of millions of dollars.

Another extremely useful resource would be an archive of source code and binaries maintained in parallel with a database of software company business records, to include marketing materials, internal memos, correspondence, board minutes, release notes, and of course the records of the development itself. Such an archive-in-depth would offer researchers and scholars a view of software development as a living activity, complete with blind alleys and breakthroughs, rather than as an orderly procession of results. There is some urgency to collecting these business records, since they tend to disappear when companies are acquired and lose their separate identities. An effort should be made to locate these ephemeral materials and arrange for their donation while they are still available.

For both these projects it will be important to concentrate on companies formed between 1960 and the mid-1970's, since this period threatens to become a historical "lost continent" in context. Until 1960, software development was primarily the province of hardware companies and of the IBM user group SHARE, and these entities had policies and resources in place that resulted in relatively complete documentation. After 1975, on the other hand, software development for microcomputers aroused considerable public interest, and several books—notably Doug Carlston's *Software People*—were written about the founders and activities of micro software companies. The years in between are a largely unknown quantity; yet, during this period, entrepreneurs developed the business models for selling software that have become today's industry standards. As one example, the concept of granting the user a license to use software—rather than selling the code outright—was first implemented by Informatics Corporation in 1967. Today, despite certain obvious limitations, this remains the most important method for granting title to commercial software.

This lack of knowledge can be remedied most immediately by gathering the oral histories of software entrepreneurs. Interviews with early developers would be especially valuable in bolstering our understanding of the entrepreneurial personality and of the challenges involved in creating unprecedented markets. Complete versions of these interviews would be made available to business and industry historians; edited versions would be offered to the general public as part of the software history exhibit. What Web entrepreneur, tired of hearing that no one can make money on the Net, wouldn't rejoice at hearing software developers of the 1960s recall being told that they couldn't make money selling software?

To sum up, much remains to be done to preserve the history of software; and, as with all computer history, this involves work best done quickly! I am drafting guidelines for the establishment and protocols of a Software History Center and would especially appreciate hearing from anyone involved in the development of mainframe, mini or early micro software. Suggestions concerning the Center's operations and priorities are also welcome. I can be reached at luanne@softwarehistory.org or through the CHAC at engine@chac.org.

## THANKS TO....

Christine Comaford for advice and hospitality.

Fred Davis and Sylvia Paull for their ongoing advice and support.

Dolby Laboratories and Scott Robinson for Sun and North Star hardware [see next issue]

Network General Corporation, Stanley Parent, and Robert Praetorius for their donations.

Dag Spicer for the fine portrait of Cuthbert Hurd.

## NEXT ISSUE / COVER ART

At least one interview, maybe two.... an article about one of California's earliest mainframes ....more on the accelerating museum activity in the Bay Area, and it looks like we may need multiple book reviews too. Computer history is taking off big-time (oh joy) and we're keeping up as best we can!

*Cover:* A Bachrach portrait of Cuthbert Hurd, from the IBM Archives.

# MICROPROCESSOR AND MICROCOMPUTER:
## Invention and Evolution

STANLEY MAZOR,
Senior Member, IEEE

In 1970 Scientific American showed a "computer chip" with 400 gates, but an entire CPU would need about 10 times that number. Because of the ever-growing density of LSI we thought a single chip CPU might be accomplished by 1980. However, by scaling down requirements and using a few other "tricks" described here, we developed two "microcomputers" 10 years ahead of "schedule."

### INTRODUCTION

Intel's first microcomputer ad appeared in November 1971 and boldly stated: "Announcing a new era in integrated electronics." In that same month the company delivered its first microcomputer chipset, the MCS[Micro Computer System]-4, which emphasized low cost. The considerably more versatile MCS-8 was delivered only five months later, in April 1972. These CPU chips were powerful alternatives to random logic, selling in quantity for less than $100 each, but aimed at two very different markets; the MCS-4 would eventually lead to the under-$1 controller, while the MCS-8 was the ancestral engine of the now ubiquitous personal computer (PC).

Over the ensuing 20 years, the prophecy of "a new era" has been fulfilled. But in 1971, advertising copy aside, we were not thinking so grandly. Our challenge was to scale down a general-purpose computer to the size of a single chip. Imagine for comparison that the only passenger vehicle in existence is an eight-seat van costing $50,000. At first it would be hard to imagine the specifications for a $1000 version of this vehicle. Some ideas to consider would be drastic reductions in capacity, speed, or range. The golf cart might be the result. However, it would not be easy to envision scaling down a van into a golf cart, if such a vehicle was unknown at the time. It might also be difficult to anticipate the potential usefulness and marketability of the golf cart yet to be invented.

The scalability of a computer depends on intended use. Fortunately for us, our first customer wanted us to build a "microcomputer" with speed and memory size appropriate to a desktop calculator. Considered as "a computer," this device was not very capable, and some probably said that we set the computer industry back 10 years. We took the far more optimistic view that we were moving the LSI world ahead by 10 years. It is obvious in hindsight that we were simply moving the world, toward eventualities we never could have foreseen.

### INTEL MCS-4 4-bit CHIP SET

In 1969 Intel, then a memory chip company, accepted a commission from Busicom of Japan to design eight custom LSI chips for a desktop calculator, each with a special function—keyboard, printer, display, serial arithmetic, control, etc. I joined Ted Hoff's Application Research group in September 1969 to help design the calculator. I had just spent 3 years helping to design a decimal computer at Fairchild R&D. I will share my memories of this period at Intel.

With only two designers on staff, Intel needed to minimize the number of chip designs. Ted chose a programmed computer solution using only one complex logic chip (CPU) and two memory chips; memory chips are repetitive, and were already within Intel's prior art.

A year later Intel designers implemented a 4-bit computer on three LSI chips—CPU, ROM, and RAM—in 16-pin packages, the only ones then available to us. Reducing the data word to a 4-bit BCD digit, a compromise between l-bit serial calculator chips and the 16 bits of conventional computers, made the CPU die size practical at roughly 2200 transistors. Limited pin count forced us to time-multiplex a 4-bit bus; but the small bus used fewer connections and simplified the printed circuit board. On the other hand, the multiplexing logic required address registers in the specialized ROM/RAM memories and increased their chip area. Increasing transistor count to save chip connections was a novel idea, directly opposite to the principle we had learned in school, but one which gradually became quite popular.

## MCS-4 Features

The original MCS-4 chipset comprised four chips: The 4001 ROM, the 4002 RAM, the 4003 shift register, and the 4004 microprocessor.

Conventional calculators of Busicom's day required specialized custom chips for keyboard, display, and printer control. The MCS-4 combines all control logic in firmware. A single ROM chip design is customized during manufacturing, with a mask corresponding to a customer's particular program; the mask also configures each ROM port bit as an input or output.

The CPU's 12-bit program counter addresses as few as one or as many as 16 ROM chips. The Busicom desktop calculator used four.

## ROM Chip (4001)

The 4001 is a 256 bit x 8 Read Only Memory (2 Kb ROM) with a 4-bit I/O port. It features an integrated address register, an output data register, multiplexors, and control and timing logic.

## RAM Chip (4002)

The 4002 is an 80 bit x 4 Random Access Memory (320-bit RAM) with a 4-bit output port, organized in response to the Busicom design's use of 16-digit decimal floating point numbers. Each 80-bit word comprises 20 4-bit digits: 16 for the fraction, two for the exponent, and two for signs and control. Each chip stores four 80-bit words and includes an output port.

The use of three-transistor dynamic memory cells made the RAM chip feasible. Data integrity is sustained by a built-in counter which refreshes cell contents during instruction fetch cycles, when RAM data is not accessed. The 4002 RAM chip has register, multiplexor and logic resources similar to those of the 4001.

## Input/Output Ports and the 4003

The integrated 4-bit ports of the 4001 ROM and 4002 RAM chips allowed direct connection of I/O devices. This conserved chip count and made use of existing power/clock pins. To activate an output, a program selected a RAM or ROM chip through an index register, and sent four data bits from the CPU accumulator to the selected output port. The desk calculator display, keyboard, and printer were connected to these ports. Keyboard scanning, decoding, and debouncing were done under program control of the I/O ports: printer and display refresh was done in firmware. A small shift register, the 4003, was used for output port expansion. External transistors and diodes performed amplification and isolation.

## CPU Microprocessor (4004)

The 4004 is a 4-bit CPU chip with 16 x 4-bit index registers, 45 one- and two-byte instructions, a four-level subroutine address stack, and a 12-bit program counter giving 4,096 possible addresses.

In the calculator application, each user keystroke caused thousands of CPU instructions to be executed from ROM. We wrote many subroutines which operated on 16-digit numbers stored in RAM. The 4004 can execute approximately 60,000 instructions/second, and a 10-byte loop for digit serial addition took about 80 $\mu$s/digit. In this add routine a CPU index register would address each of the 16 digits stored in the RAM memory; the program would bring one digit at a time into the accumulator register to perform arithmetic; finally, a Decrement and Jump instruction would index to the next RAM location. The resulting speed was comparable to that of an IBM 1620 computer which sold in 1960—eleven years earlier—for $100,000.

The MCS-4 was distinct from most contemporary computers in its implementation of separate program and data memories, with application firmware permanently stored in ROM and data transient in RAM. In a typical minicomputer of the day, such as the DEC PDP-8 or HP 2114, both program and data would be resident in RAM and the calling program's return address would be saved at the top of the subroutine. MCS-4 routines in program ROM, unable to be written to, forced a major change in subroutine linkage.

Ted Hoff and I had programmed Burroughs computers and the IBM 1620, which used stacks. We used this experience with large-scale computers to install a push-down stack inside the 4004 CPU, accommodating up to three return addresses. Ultimately this limited depth of four levels—all we could squeeze onto this small chip—was frustrating for programmers; succeeding generations of Intel processors went to eight or more levels in the 8008, 4040, and 8048. Today's computers have stacks of many megabytes, but their use is very similar to that of the 4004. Another Intel innovation, drawing on prior experience in memory design, was the use of dynamic RAM cells inside the CPU for the 64-bit index register array and 48-bit program counter/stack array. Briefly, the highlights of MCS-4 architecture are time division multiplexing of the 4-bit bus, on-chip dynamic RAM memory, and the CPU address stack.

One more interesting feature of the MCS-4, not used in "conventional" computers, is distributed decoding of instructions. The ROM/RAM chips watched the bus, and locally decoded port instructions as they arrived from the ROM. This eliminated the need for separate signal lines between the CPU and the I/O ports, and saved CPU logic.

## MCS-4 Applications

Busicom of Japan produced several calculator models using this chipset. I had added 2 instructions in the CPU set, FIN and JIN, to implement an interpreter to Fetch from ROM and Jump INdirect. We wrote a 20-byte interpreter, and implemented a pseudo-machine with 16-digit fixed point operations for the calculator.

The smallest MCS-4 system would contain two chips, a CPU and a ROM. A typical calculator had 4 ROM's and a RAM, giving five I/O ports. A fully loaded system could have 16 ROM and 16 RAM chips, and obviously a plethora of I/O ports. The MCS-4 was typically used in digital scales, taxi meters, gas pumps, traffic lights, elevator controllers, vending machines, and medical instruments. Such breadth of application essentially assured the future of the microprocessor as an embedded controller.

Ted Hoff and I made the original proposal for the MCS-4 and did the feasibility study for the first calculator. Federico Faggin did the logic and circuit design and implemented the layout. Masatoshi Shima of Busicom wrote most of the calculator firmware, later joined Intel, and designed the 8080. The Intel patent on the MCS-4 (Hoff, Faggin, Mazor) has 17 claims, but the single-chip processor is not claimed as an invention.

Intel supported the MCS-4 with a cross-assembler and later with a stand-alone development system, the Intellec "blue box." This in turn brought attention to the marketing efforts of H. Smith. R. Graham and Ed Gelbach.

The MCS-4 evolved into the single-chip 8048 and 8051 microcomputers which emphasized small size and low cost. These, along with a variety of other manufacturers' parts, have become the under-$1 single-chip microcontrollers now pervasive—and almost invisible—in toys, automobiles, and appliances.

## INTEL 8008 MICROPROCESSOR (MCS-8)

An early Intel customer, Computer Terminal Corporation (now known as Datapoint,) produced a low cost bit-serial computer for which Intel supplied a custom 512-bit shift register memory chip. This CTC 2200 computer had an 8-bit CPU, implemented with bipolar TTL MSI and including about 50 data processing instructions. In December 1969, CTC's Victor Poor asked me about building a 4x16 stack chip using our bipolar 64-bit memory; based upon our progress with the MCS-4, I proposed an 8-bit parallel single chip CPU, to be called the Intel 1201. Ted Hoff and I suspected that a substantial reduction in chip count could be achieved through use of MOS technology. After some stops and starts this custom chip design was completed, but it was never used by Datapoint and instead became a standard Intel product, which marketing renamed the 8008—twice a 4004.

Although the arithmetic unit and registers were twice as large as in the MCS-4, we expected that the control logic could be about the same if we deleted a few Datapoint defined instructions. Unlike the MCS-4 with split memory address space, the 8008 had a common address space for program and data. The symmetric and regular instruction set was attractive. However, the only memory addressing was indirect through the High-Low (HL) register pair.

The 8008 CPU had six 8-bit general purpose registers (B, C, D, E, H, L) and an 8-bit accumulator. The push-down program counter stack had eight levels. Both of these register arrays were implemented with dynamic memory cells and the CPU had built-in "hidden" refresh during instruction fetch cycles, similar to the MCS-4.

We decided that the 8008 would utilize standard memory components, available in high volume at low cost, rather than custom ROM's and RAM's as for the MCS-4. This increased the parts count on a minimum system by requiring discrete address registers, multiplexors and I/O latches; in practice about 40 additional small chips were needed, but in larger systems the extra chip overhead could be tolerated. Using memory chips with different access times requires synchronization, and therefore ready/wait signal pins were provided to perform handshaking. Thus the 8008 demonstrated an interface signaling scheme still used, in much more sophisticated form, by today's processors.

Intel had an 18-pin package in volume production for the 1k dynamic RAM chip (1103); this allowed two more pins for the 8008 than for the MCS-4, but we still had to time-multiplex an 8-bit bus. By reducing the program counter width to 14 bits we saved two package pins. The jump instruction contained a 16-bit address, but two of the bits were ignored. The 8008 could address 16 Kbytes of memory, which at the time seemed enormous.

Most of the 8008 instruction set carried over from Datapoint's definition by H. Pyle and V. Poor. Hoff and I wrote the specification for the 8008 single chip CPU. Hal Feeney did the chip design under Faggin's supervision; I did the logic simulation for Feeney. Les Vadasz was our overall manager.

Some have wondered why addresses in the 8008 were stored "backward"—little-end-first, with the low-order byte of a two-byte address stored in the lower addressed memory location. I (regrettably) specified this ordering as part of the JUMP instruction format, in the interest of compatibility with the CTC 2200, whose original processor was bit-serial and whose addresses would be stored low to high bit in the machine code (bit-backward). Other computer makers organize the addresses with the "big end" first. The lack of standardization has been a problem in the industry.

## 8008 Applications

The 8008 proved a popular chip for applications, partly because Intel could supply compatible Electrically Programmed ROM's (EPROM) which would make it easier for customers to experiment with software development. Therefore, the microprocessor in turn gave a significant boost to Intel's memory component business.

One of the first users of the 8008 was Seiko in Japan for a sophisticated scientific calculator. Other uses included business machines and a variety of general purpose computers. Sandy Goldstein wrote a cross-assembler; Gary Kildall of Digital Research created PL/M-8 and then CP/M, the famous operating system instrumental in the development of Microsoft's DOS.

Intel did not apply for a patent on the 8008 because we didn't invent the instruction set, and because it was basically integration of a discrete CPU design. Datapoint contracted with Texas Instruments in 1970 to get a second source for this chip. Engineers at TI have said that they thought it would take 3 chips to implement the CPU in metal-gate, but on the strength of Intel's 1-chip proposal, they proceeded to design a single-chip CPU. They did patent their design, but never went into production, and I wonder why. It is ironic that Datapoint ultimately competed in the marketplace with PC products based upon *their own,* Datapoint defined, architecture!

## 8080 MICROPROCESSOR: More and "No More"

After about one year of experience with programming the 8008 CPU chip, we had logged several requests for enhancements from our users. In 1972 Faggin convinced Vadasz to fund a follow-on chip, the 8080, which would convert the P-MOS 8008 to the newer N-MOS technology. Speed was anticipated roughly to double without making logic changes. A short study determined that a new mask set was needed because of the incompatibility of transistor size ratios. Faggin recognized this as a chance to fix some of the 8008's shortcomings.

We evolved the 8080 specification for tenfold improvement in performance. We used the greater density to put in more logic (~4500 transistors) and do more in parallel; the on chip control logic grew by 50%. We put the stack in memory, did 16-bit operations, and improved memory addressing. Now 40-pin plastic packages were available, and the address bus and data bus could be brought out in parallel. This design also simplified the external circuitry and TTL voltage compatible signals were provided.

Deleting the on-chip stack saved chip area and offered unlimited stack size to the user. I defined the stack as growing downward from the high end of memory; this facilitated indexing into the stack and simplified displaying the stack. This was abandoned on the 8086.

In the 8080, three registers were arranged as pairs of 8-bit, to provide 16-bit data handling. To the old HL (High/Low) register pair which was the only way to address memory in the 8008, we added register pairs designated as BC and DE, which could be used with direct memory addressing instructions. Several specialized instructions were added for the HL register pair: XTHL provided for exchanging the top of stack with HL; another instruction; XHLD swapped the contents of HL with the DE register pair. As these special instructions were not very symmetric, applying only to HL, we optimized their logic implementation. One of Ted Hoff's tricks was the use of an *exchange flip/flop* which designated one of the pairs as HL and the other register pair as DE. Simply toggling this flip-flop effected an apparent exchange and saved a lot of logic. Unfortunately, by mistake, the reset pin had been connected to this flip/flop, and an early 8080 user manual conceded: "after reset, the HL/DE register contents may be exchanged." Later the reset connection was cut. Lack of instruction set symmetry was a nuisance to programmers, and later CPU's with more transistors "to burn" had instruction sets that were considerably more regular.

## Shima's "No More"

My specification for the 8080 used all 256 possible opcodes. Twelfth from the bottom of my list was the aforementioned (and obscure) XTHL instruction, which would require five memory cycles to execute, and be used to pass arguments to subroutines. I carefully explained each instruction to Masatoshi Shima, the 8080 project manager under Faggin, whose patience was tested as I detailed XTHL's operation. He drew a line under this instruction, and declared: "No more."

This is why the last 12 instructions of the 8080 were never implemented. The twelve spaces in the set were filled with instructions added to the 8085 microprocessor; but the selection of those instructions was so protracted that the 8085 was almost obsolete when it was introduced, and largely missed its mark.

The 8080 CPU chip was patented by Intel (Faggin, Shima, Mazor) and has three claims; it led to the microcomputer revolution and the affordable personal computer. It was very successful in a competitive market where great new processors, including the Motorola 6800 and the MOS Technology 6502, blossomed. Shima, Faggin and Ralph Ungermann (perhaps better-known as a co-founder of Ungermann-Bass) formed Zilog, and competed with an enhanced processor, the Z-80, which was used in personal computers and went on to enduring success as an embedded controller. Intel's designers continued to meet competition by introducing the 8085 in 1976, and the 8086, a true 16-bit microcomputer containing about 30,000 transistors, in 1978. IBM's decision to use the 8088, a version of the 8086 with an 8-bit data bus, in a word processor and personal computer created enormous market momentum for Intel. Some shadow of the "truly pervasive" 8008 feature set would still be apparent in the 80186, 80286, i386 and i486, representing 15 years of unrelenting development in commodity microprocessors.

## HISTORICAL PERSPECTIVE: Technology Predictions

The promise of high-density solid state circuitry was apparent by 1959, when Holland contemplated large scale computers built with densities of 10^8 components per cubic foot. in the mid-1960's Fairchild's Gordon Moore and Robert Noyce made forecasts which assumed that the density of IC's would double every year.

By 1966 Petritz of TI was forecasting about 10 k transistors per chip for 1970 and 100 k (optimistically) by 1976. Hobbs at that time foresaw the reduced cost of arrays, predicting that CPU cost would become "negligible". Practical people predicted "entire subsystems" of about 10 k-20 k gates on a single chip, and recognized that the issues were the "number of unique part numbers and the production volume"—since, after all, only a few thousand computers were made each year. Most predictions of the day presupposed a high volume standard chip definition.

## SSI, MSI, LSI Chips

By 1968 16-bit minicomputers utilized a single printed circuit board CPU containing around 200 chips. These were medium scale integrated circuits (MSI) with ˜ 100 transistors per chip, and small scale IC's (SSI). Obviously. the more transistors that could be put on a chip, the fewer chips would be needed on a PCB. Manufacturers, trying to reduce costs, battled constantly to reduce chip count—could a CPU be built with 150, 80, perhaps 25 chips?

By 1970 there were a few attempts to build 16-bit minicomputer CPU's from multiple LSI chips, defined as 1000 transistors each. These were full 16-bit minicomputers and did not have a scaled down specification like the MCS-4, except for their physical size. They utilized 4-bit or 8-bit arithmetic and register "slices": a minimum CPU would require 8 to 12 LSI chips with about 6 to 8 different part numbers. Such projects were R&D with military sponsorship—particularly from the Air Force, which was exploring lightweight airborne minicomputers—at Raytheon and RCA.

## LSI Economics

Application of custom LSI requires very high production volume—around 100,000 systems—to justify tooling costs and attain commercial feasibility. In the early 1970's the only high volume commercial applications were calculators: almost every calculator manufacturer was designing custom, specialized LSI chips for arithmetic, printers, and keyboards. Busicom's original request was for such a chipset.

Final chip cost depends not only on tooling but on optimum die size. If a die is too small it cannot contain enough circuitry to justify a fair price. If a die is too ambitious and large, manufacturing yield will be too low and the chip will be too expensive. Worst of all, at the beginning of a complex chip project it is not easy to forecast accurate final die size. Defining standard high volume LSI chips is challenging.

In 1970, therefore, no one had defined LSI building blocks that were usable in a variety of applications. The only LSI building blocks available were memory chips. Honeywell tried to get multiple sources for a 64-bit bipolar LSI memory chip. but that was on the leading edge of bipolar technology, and not many vendors could make them. Metal-gate MOS ROM's and 200-bit shift registers were available from sources including AMI, Electronic Arrays, MOS Technology, and General Instruments. These were easy to design, with very regular structure and minimal internal wiring; each chip had around 1000 transistors and was two to five times more dense than random logic chips.

## Partitioning and Packaging

Any system implemented on a set of LSI chips must be partitioned into cost-effective packages with a reasonable number of pins. In 1970 the most common commercially available low-cost packages had only 14 pins and sold for around $1. Packages with more than 20 pins were very expensive, and 48-pin packages were then selling for around $10. Only the least expensive packages were appropriate for cost-sensitive applications such as desk calculators.

Optimization consists of maximizing the number of gates inside for each pin outside—the gate-to-pin ratio. Memory chips with 1 Kb in an 18-pin package gave an excellent gate/pin ratio of about 100:1. Each time the technology allowed a doubling of bits on a chip, only one more address pin was needed. A shift register was even better, because regardless of the number of bits added, the input/output pin count stayed constant. But if a CPU were to be built of LSI chips, it was not obvious how to break it into pieces with a small number of I/O pin connections and a high gate/pin ratio. Each time you cut an ordinary CPU into two chips, you would force hundreds of signals to cross the chip boundaries.

Each package pin also required a lot of MOS chip "real estate" for amplifiers to cope with heavy off-chip capacitive loads, and for wire bonding pads connecting the chip to the package. Placing more pins on an LSI chip not only increased cost, but diminished reliability. Hence, most early commercial LSI applications—including microprocessors—were constrained by the few leads available on IC packages, typically with 16 or 18 pins.

## Semiconductor Technology

On-chip interconnections were also a major problem. A CPU chip contains "random logic" requiring numerous interconnections. Prior to 1980 most semiconductor chips had only one layer of metal, which was reserved for global connections such as power, ground, clocks, and major busses. Local connections were made using lines of polysilicon or diffusion, of poorer quality and with higher resistance.

The silicon gate process, developed originally at Fairchild Semiconductor circa 1967, provided slightly better local interconnections and cross-overs, as well as lower capacitance, lower voltage operation, and smaller size thanks to self-aligned structures. This technology was key for microprocessor development at Intel, which used it to implement the 8008 chip for Datapoint. In contrast, TI's contemporary metal gate MOS technology used about twice the area of silicon for a similar chip.

Silicon gate P-MOS needed a 14 V supply, often biased between +5 V and -9 V to give pseudo-TTL compatibility—a relatively high voltage in context of the stringent power budget facing the circuit designer. Small IC packages cannot dissipate more than 1/2 W of power through ambient air cooling. The compromise was to use dynamic logic and reduce temperature with low duty cycles. In 1996, power dissipation is still a major concern of commercial system design; it has been one of the driving factors toward 1.5-2.5 V technology, another factor being the popularity of battery operation.

Dynamic operation of P-MOS circuits required two phases; first the circuit was precharged using an on-chip amplifier, and then it was conditionally discharged, based upon logic decisions. Previously, "bootstrap" amplifiers were built using the gate "overlap capacitance" as part of the circuit. However, silicon gate self-aligned geometry eliminated this capacitor. Federico Faggin invented a new and efficient bootstrap amplifier as part of his early circuit design of Intel's chips. This reduced overall power dissipation.

The dynamic RAM cell, a crucial insight of Ted Hoff's, also made micros feasible. The memory bit is stored as a charge on a small capacitor, usually integrated into a three-transistor memory cell. After every 5 ms, an external "refresh" circuit reads, tests, and restores the charge. Static memory cells, which have some theoretical advantages, were impractical for use inside the CPU because they required twice the chip area and used much more power. The use of dynamic RAM for index registers and stack inside the CPU was essential in the design of the 4004 and 8008.

## CAD Tools

"Home grown" tools for circuit analysis had been a tradition among computer makers since the mid-1960's. Hoff and I developed a transient analysis program, PULS, to help with MOS circuit design; Intel designers later used it to help them achieve the desired AC/DC performance. Intel's Dov Frohman, who invented the EPROM (he didn't call it a FROM), provided the transistor model. Hoff wrote our first logic simulator for the PDP-8; later I used a commercial tool, Applicon, for the PDP-10. I abused the DEC macro assembler to get the first MCS-4 code assembled and into the 4001 ROM bit map. We also used it to develop the early calculator firmware. These CAD tools allowed our designers to catch design errors early and were qualitative to Intel's success. Several CAD companies in Silicon Valley were spawned from these in-house CAD groups.

## Evolution of the Micro- Prefix

In the mid-1960's, small computers in the marketplace were referred to as "minicomputers." Some computers used microprogramming stored in ROM: the inner part of such a computer was called an "engine" or "microengine" or "microprocessor." Somewhat later, Fairchild had a logic family called $\mu$-logic, and the prefix carried over generically to "micrologic" in IC's. (Intel did not use the Greek letter; we avoided references to Fairchild's product line since many Intel employees had come from Fairchild.) The earliest mention of the "computer on a chip" is probably from 1968.

In 1970, *Scientific American* featured a "Computer on a Chip," with 400 gates. At that time a "microcomputer," which had not yet been built, was defined in theory as a computer considerably smaller than a minicomputer and possibly using ROM for program storage. When the commercial "microcomputer" arrived on the market in 1975 it was understood to be a computer whose CPU was a "microprocessor," in many cases the Intel 8008 or

8080. More recently, and by extension, the terms "nanocomputer" and "picocomputer" have been used by computer engineers discussing theoretical devices of extremely small size and performance of computers. Today, the single-chip central processor unit equipped to use off-chip memory (such as the 4004) is commonly called a microprocessor or microcomputer; self-contained single-chip computers are often called microcontrollers.

### SUMMARY

Integrated circuit technology has been evolving along a predictable path for the past 30 years. The creation of a computer on a chip with LSI chips, which had fewer than 20,000 transistors per die, was problematical; most such work focused on partitioning 16-bit computers into multiple chips, but few of these projects were successful. Early Intel microprocessors succeeded because they were computers scaled down to the point that all appropriate random logic could be concentrated on a single die; this entailed a decrease in word size from 16-bit to 4-bit. When the densities reached 200 k+ transistors per chip and word size grew once more to 8-bit, 16-bit and even 32-bit, microprocessors became the dominant computer technology.



## UP ON THE RIDGE: The Story of a Workstation Pioneer

### An Interview with John Sell

*KC:* I did read Kunze's book[1], which has an odd slant on the technical side, because he was concerned more with the money angle—I mean, it's a book about venture capital—

*JS:* And he's not a technical guy. I didn't think it was odd. He was generally favorable.

*KC:* This is true. He thought a great deal of the company and of some of its accomplishments, and to begin with I wanted to focus on those accomplishments. What were you trying to do and what did you succeed in doing?

*JS:* Well, we saw a way to build machines with much higher performance and potentially much lower cost than our competition at that time, and from the wide range of computers we could have built, we were determined to focus on scientific graphics, rather than on big banking or database-handling machines, or analogs. In the early days we said we were building a "personal mainframe," and I believe we coined the term, which would nowadays be applied to a superpowerful desktop computer or a workstation. Now, we didn't coin the acronym RISC, and of course neither did the people at IBM with the 801; I assume it originated with Hennessey and Patterson, probably about 1982, after Ridge started.

*KC:* So Ridge started in—

*JS:* May 1980.

*KC:* And when were you able to bring out your prototype?

*JS:* I know that we had the first prototype running, in a card cage sitting on a cart so we could wheel it around to different workbenches, probably in the fall of 1981. We were self-funded at that time, but on the strength of that, we went outside for our first venture capital and closed the deal just before Christmas of that year. I think we delivered the first prototype for beta test in the spring of 1983,

[1] Robert J. Kunze, *Nothing Ventured: The Perils And Payoffs Of The Great American Venture Capital Game,* HarperBusiness, New York, c. 1990.

and we showed finished systems at Comdex in the fall of '83, so about two and a half years after we started. Then there was a lot of software development.

*KC:* While you were developing the prototype, what did you see as the competition?

*JS:* When we started, the DEC VAX 11/780 was available and the smaller version—the 11/750—was just coming out. Apollo, which of course is now part of HP, came out with systems which were physically similar to ours, file-cabinet-sized deskside systems—I think the first ones were 68000 or 68010-based, but by the time we actually came out the door, they offered systems built around the 68020. So that was the range of the competition. In sales situations we basically saw DEC most often. Some people took a chance on a new company and bought our systems simply to save floor space—we gave them a little box rather than something that took up half a room—and of course almost everybody bought them because of price. Ridge offered great performance and they could buy computing power from us at a fifth, or maybe less, the cost per MIPS of an equivalent VAX.

*KC:* In fact the Ridge computers offered higher performance in absolute terms than most of the competition.

*JS:* Right. If I remember correctly, the original Whetstone benchmark was normalized to the VAX 11/780 as 1, with whatever compilers they used. On that benchmark, the first Ridge system delivered in 1983 was between 1.5 and 2. Depending on what you were running, a Ridge could be about twice as fast as the VAX.

*KC:* Was this done with custom processors?

*JS:* It was done with a Ridge-developed custom architecture. The first system was built out of PALs, 7400-F series logic and other discrete components, and of course we included floating-point data paths. The whole Ridge CPU—kind of the CPU-memory-I/O interface—was a pair of circuit boards. Now, later on in the development of Ridge, Sun came along, and their earliest systems were 68000-based with Weitek floating-point chips. The Ridge had two circuit boards and the Sun had one, but the total size of the implementation was about the same; and our whole CPU subsystem cost less than what Sun had to pay Weitek for a pair of floating-point chips, yet it outperformed them. The point is that VLSI, in the early 1980s, really wasn't all that dense or cost-effective, especially when the more complex 68000 architecture was compared to a RISC architecture.

Of course that changed and in the late 1980s—I'm leaping ahead here—the founders of Ridge, including myself, perceived that the density and speed of VLSI logic had finally improved such that, if we didn't implement systems that way, we wouldn't be competitive. We didn't have the resources by ourselves—for lots of reasons, but briefly, Ridge had not been successful enough to afford the design tools, and designers, and overall costs of full custom VLSI. We could no longer do it the way we had been, and we wanted to sell the company, and we disagreed over how to do that with the venture capitalists, whom we ultimately left.

*KC:* Had you intended to start with discrete logic at a time when discrete logic was most cost-effective, and then down the road go to VLSI—

*JS:* That was our plan, and the last Ridge systems which were built were in fact gate array implementations. Rather than go to full custom VLSI, we used the same Fujitsu gate arrays that the first Sun SPARC was built out of, and we had prototypes built with those ASICs that were up and running slightly before Sun had SPARC systems running. We both ran at the same 16-MHz clock rate, because in terms of taking it to the goal registers through an ALU data path, that was about the best you could do with those ASICs; but our architecture and our implementation were superior in terms of how many clocks certain instructions took, how the memory interface worked, how we handled things like allowing cache access. We used a technique for handling cache misses that SPARC didn't use in its first implementation, that allowed us to beat the earliest SPARC by about 50 percent, even though we were both running at the same clock rate on a wide range of benchmarks. By then, on the different benchmarks that people would give both Ridge and Sun, we were beating them by 50 percent or even sometimes double, because we were taking in effect fewer clocks per operation.

SPARC, MIPS, and the Ridge architecture all had their differences, but they all have many attributes of RISC—lots of registers, a fairly simple instruction set, the ability to perform most instructions at one per clock and to pipeline them easily. Essential elements of RISC are common to many of the architectures.

*KC:* Okay, but wait a minute. All of this was built out of discrete logic, and that inevitably meant longer signal paths.

*JS:* The one I was just referring to, the gate array implementation, had all its critical paths inside the gate array, so yeah, there was discrete logic, but for things like bus drivers and SRAM for a large external cache memory—which both [Ridge and Sun] had, because building a cache memory on board out of a gate array didn't make a great deal of sense. Certainly fetching from the cache was an issue, going off-chip to the cache, but we mitigated that with careful board layout that put the cache and CPU chips in their own corner of the board, and let things like I/O communications go across longer buses through the box to other parts of the board. You know that even relatively modern HP Precision architecture machines have large external caches that are separate from their custom ALU chips, and at today's hundreds-of-megahertz clock-speeds that's a much more serious issue than it was. Back when we were running at 16 MHz, the extra few nanoseconds that the PC traces entailed weren't really significant.

*KC:* Even so, there was some fairly sophisticated engineering going on here, and who was doing it?

*JS:* Well, actually five of us developed the first system; myself, Hugh Martin, Dave Folger, Ed Basart—incidentally, that's the correct spelling of Ed's name, it's been wrong in other sources—and Ron Kolb. All of us except Dave Folger were from HP; Ron and Hugh worked for me there and Ed was a colleague who had worked on the operating system for the HP 3000. (For the record I should also mention Neil Wilhelm, who didn't stay with the company, he left after the first couple of months.) But the five of us developed and built the first system, had the prototype up and running with the beginnings of its operating system. We reached that point while we had funded the company ourselves through that first year and a half, invested $500,000 and worked without a salary. We started out in a shed behind Ed's house, and while we were still self-funded, we hired Marj Kondo, who began as our receptionist, but later became our administrative person who ran the personnel department, managed the facilities, and did a lot of other things.

The company grew, and we wanted to retain as much control as we could—because none of us were industry veterans, if you will. When we started the company, I was 29 years old, and most of us were within a couple of years of that age one way or the other. We didn't need money to get started; we felt we'd be in a better position if we proved that we could produce something before tried to raise money, and that's what we did. But after the first few months we moved to an office complex, started paying rent, and went outside for venture capital. We approached Arthur Rock initially because we'd read about him, basically, and he sounded like an individual rather than just part of a firm managing other people's money; he seemed to be someone who would actually look at an opportunity and place his own bet, whereas the larger venture capital firms are more like banks—of course they're not banks, but they're hired to manage money.

*KC:* They play by hard and fast rules.

*JS:* At least harder and faster, and they might have more trouble seeing the uniqueness in a unique situation. Arthur, on the other hand, was very interested. He turned out to believe in what's now called RISC as a concept, he liked our implementation of it, and he agreed to become an investor. Then he approached Hambrecht & Quist, with whom he had a strong relationship at the time, and asked them if they wanted to get involved, and ultimately the two of them split the investment that was made; exact percentages aren't really relevant, but through the first two or three rounds of financing, in successive years, the founders actually still owned a majority—which is altogether unusual for a hardware company. And it was that way because we had actually produced something first ourselves, rather than raising money and then starting to do something.

*KC:* Meanwhile, you were doing the hardware and software development concurrently. Were the same people doing both?

*JS:* No, Hugh and Ron and myself were the hardware people, and Dave Folger and Ed Basart developed the software. We pretty much divided the work up that way.

*KC:* What about the operating system? You made a conscious decision not to go with UNIX.

*JS:* And that was a mistake.

*KC:* Well, but it was a proprietary operating system. What was there about it that you felt was going to offer the same degree of advantage, of leverage, that the hardware did?

*JS:* I hate to beg off on that by saying "Gee, that wasn't me," but to some extent, that was what happened. We hardware guys said that we could make an architecture that for a given amount of logic was much more efficient, and we did, and our software founders thought that they could do the same thing on the software side. At the time, without benefit of hindsight, it wasn't quite as obviously necessary to follow some kind of developing industry standard. Our main competition at that time was DEC, which relied on proprietary operating systems, HP used proprietary systems for their minis and weren't our competitors then anyway, and the same was true of Burroughs, IBM, Data General, NCR—you name it. There was no Sun when we made that decision, and there were no HP workstations. UNIX in 1980 was a system with its roots in the academic community that hadn't caught on nearly to the extent that it would later—it was on its way up, but it hadn't grown into prominence. The other thing was, of course, the complication of "Well, but if you're UNIX, which UNIX," with half the prospects wanting us to be [BSD] 4.3 and half wanting us to be [AT&T] System V. We confronted this from the standpoint of having five decision-makers and limited development resources, and thought that if we implemented a system which did everything we needed it to do, without worrying about complying with and chasing a particular standard, we could do a lot more with the resources at hand. That part was probably true. Now we can look back, while UNIX is fading as NT takes over, and understand the historical accident of relying on a proprietary OS during the exact heyday of UNIX and UNIX compatibility.

Now lack of compatibility was not the Achilles' heel, no one thing killed Ridge, but certainly we were very UNIX-like and underestimated—or took too long to understand—people's annoyance with little things like "Gee, the time isn't reported in exactly the same format." We tried to be UNIX-like consciously so we'd be familiar to people, and we thought we'd done our job by incorporating the best features of both [BSD and System V,] but even the smallest differences annoyed our customers, differences that we tended to consider trivial or cosmetic. It might not have been why they did or didn't make a purchase decision, but it was definitely enough to hold Ridge back. During this period, Sun emerged as a company and was gaining momentum and market share, and we ultimately followed their lead, which certainly was the best thing to do in terms of UNIX systems—even though Ridge was then a bigger company.

*KC:* How big?

*JS:* At its peak Ridge had about $18 million in sales. Of course those were 1985 dollars, so it meant a little more than it does now. It had about a hundred and fifty people.

*KC:* How many units do you think you ultimately put into the field?

*JS:* Ridge itself manufactured and sold about six hundred, as I remember. We also licensed the architecture and the technology to Groupe Bull in France, and they bought a few from us, but they manufactured their own, including several hundred during the time of Ridge's existence. They had the right to continue manufacture without having to continue royalty payments in the event that Ridge were to go out of business, so after Ridge did liquidate, Groupe Bull made some number of machines, but I don't know how many. If you asked me to guess at an absolute total, I would say it was somewhat more than a thousand. These are small numbers by today's standards, but they were of some significance at the time.

*KC:* Sounds just on the edge of being a real collectible to me—nowadays.

*JS:* True.

*KC:* One of the things that Kunze mentioned in his book, without being too explicit about what it meant, was the term *Northface.*

*JS:* Well, I don't believe that we ever actually named a system that. The first Ridge system didn't have a name, because the five of us were just building the thing. All the models after the very first one were named after ski runs. I think Kunze mentioned that in his book, and he may have grabbed "Northface," but it's not a name I remember. The first production system with discrete logic was called Waterfall, and the second one Headwall, and the final system with the gate array logic was called Sunrise.

That decision had one amusing result, because Sun's internal name for its first SPARC developments was Sunrise also, and of course we didn't know that. We had chartered a bus and taken a weekend trip to Monterey, and we reserved a private room in a restaurant to have a celebration dinner for our direct design team—about twenty-six people—because we finally had that system up and running, and we handed out Sunrise t-shirts with the Ridge logo on them. Somebody who worked for Sun was in the restaurant, stopped one of us on the way to the bathroom, and asked, "Are you guys the Sunrise team?" thinking we were Sun employees.

*KC:* Now, when Ridge delivered successively more powerful systems, were they field-upgradable?

*JS:* Yes, absolutely. Every Ridge system ever sold could be upgraded from Waterfall to Headwall and then the Sunrise CPU. In fact, almost all of the customer machines were upgraded to Headwall level. The Sunrise CPU was never actually sold—there were prototypes running, but it wasn't marketed before the company was liquidated. Waterfall and Headwall were built to use the same I/O and memory systems and the same backplane.

*KC:* So you bought the box, and you had the box.

*JS:* Once you had the box with its backplane, you could upgrade by plugging in CPU cards, memory cards, and I/O cards. There were two sizes of box, but they had compatible internal framework, which stayed essentially the same, although we changed the cosmetic skins. With either size of Ridge box you could upgrade the disk drives and other peripherals while you kept, for example, your power supply. Of course it was physically larger than today's tower desktop PC, but it was very similar in that it depended on standard form factors, so that everything could be upgraded with modules at your location, just like your PC today.

*KC:* What kind of memory were you using? 256K SIMMs, like a Sun?

*JS:* The first prototypes, back in 1980 and 1981, probably used 16-kilobit chips because that was what we could get. The first machines sold [in 1983] had 64-kilobit chips, and those were replaced very quickly with 256-kilobit chips. Towards the end we were using 1-megabit chips. These were all DRAMs, and they were all mounted on circuit boards. The circuit board form factor in all of the Ridge systems was quite large, about 14"x15" or 15"x15". In the Sunrise prototype system, the one with the gate arrays, one circuit board was the complete system, except for the memory array cards.

*KC:* So that was one way in which you guys were definitely traceable back to an HP heritage. You weren't scared of building big circuit boards.

*JS:* That's right, although one of our biggest problems initially, just in the mechanics of the early system, was finding good people to do hand-routing of the circuit boards, and then later, we finally found automated tools that could do a decent enough job. We had to find good people who could actually wave-solder the darned things. It was an ordeal going through fabrication houses finding good ones— they all used the same equipment, but a lot of them didn't maintain it very well, and it's amazing how poor their results were. We finally contracted with Solectron, which was quite small at the time—of course it's very large now here in the Valley—but were the first people we found who could actually maintain the quality of their wave-solder machines so we could get the boards done.

Our relationship with Solectron actually allowed us to do the sensible thing for the size company we were. I mentioned that we hit a peak of over 150 employees, when we were stuffing all the circuit boards, building our own cables, what have you. We then gradually reduced our in-house manufacturing force while we started contracting all that out, most of it to Solectron to actually build the systems, so all we did was assemble them and burn them in.

*KC:* Right, but you had put yourself in a situation where not only your own stuff but your second-tier stuff had to be very high quality.

*JS:* We were actually very proud of that, because it worked. The Ridge systems had extremely good reliability. In fact, we made lots of money off maintenance because we charged maintenance fees that were competitive in the industry, but we didn't have to do as much screwdriver-twisting as a lot of other companies. We had sales offices in the Midwest, Chicago, New York, and New England; and when the company liquidated, at least one person in each of those offices bought up spare parts and kept servicing machines in their region on a contract basis. Some of those contractors may survive to this day as a side business—I know just hearing from them that as recently as within the last couple of years, there were still Ridge machines in operation, and for the most part they didn't need attention. But we were very careful about analyzing the quality and the reliability of the subassemblies we bought. That was kind of an HP area.

*KC:* Definitely. Now if somebody had, let us say, a typical Ridge system at the time of the full flower of the company, what did they have? Was it a desktop box or a deskside box?

*JS:* They were deskside boxes in two sizes. The earlier one was about the size of a dishwasher, precisely desk height and desk depth, and almost as wide as it was deep. The smaller system was a little smaller than a two-drawer file cabinet. It was still desk depth to accommodate all the same peripherals and cards, but it was quite a bit narrower and lower so it could fit under a typical 29"-high work table.

*KC:* Like a Xerox Alto?

*JS:* Yeah. Or the very largest sort of tower servers that you see for sale now, say, with four Pentiums in them.

*KC:* Gotcha. Now this sort of under-the-desk unit, how much room would that have for internal disks?

*JS:* The earliest systems had, I think, 14" Priam disks, but we designed the original package so you could also use 8" peripherals, and we quickly standardized around the eight-inch form factor. You could put, I believe, on the order of four such things in the large box but only one or two in the small box, and they actually were mounted on their sides rather than stacked flat. Some systems had cartridge tape units for backup, or a floppy—there weren't many, because in those days the only floppies around were those big giant ones—

*KC:* Eight-inch floppies. And then would this have somebody's standard terminal hooked up out the back, or what?

*JS:* Yeah, we supported—well, the boxes had varying numbers of interfaces, but the minimum would be RS-232 and RS-422 ports to connect up to terminals. We also had a couple of different industry standard line printer interfaces, Dataproducts and some of the others. Then, not with the very first systems, but early on, an Ethernet interface was supplied with every system. One card had all of the standard I/O on it, and also interfaced to the internal disk drives. Then we took the same enclosures that we used for the systems, gave them different front-panel skins and sold them to be filled up with more disk drives, or nine-track reel-to-reel tape systems. All of this sounds archaic at this point, but people wanted big reels of mag tape to do backups.

*KC:* But it also sounds as if by, say, 1985 you had a pretty comprehensive product line?

*JS:* We had to if we were going to sell anything at all. All the disk drives were SCSI by that time—I think maybe always—and towards the end we were moving to SCSI-II. It's amazing looking back, but the drives we were using for the bulk of the time were Fujitsu half-gigabyte 8" disks, four or five inches high, and they were really state-of-the-art and very cost-effective. Now you get a little hip-pocket thing—

*KC:* The other thing about those rack-mount half-gig drives is, they were distinctly "Don't drop *that* on your foot, then." A Fujitsu Eagle is a massively heavy thing.

*JS:* There were two different Fujitsu disks, and I think I remember one being a 10" disk and the other an 8". Maybe they were both called Eagles. When you bought the Ridge companion enclosure, you could fill it if you wanted to with the physically bigger drive, which was a gold-anodized aluminum sealed unit. I don't remember whether the bigger one had any more *capacity*—they may both have been half-gig—but I know that the 8" form was half-gig, because that's the one we mostly used. The smaller drives, the 8" ones, weren't too bad, but the larger one really did seem to weigh a ton. One nice thing, though, was that those disks had very fast seek times, for their day or even for today.

*KC:* That's true.

*JS:* And as you probably know, seek times have almost as much to do with better performance as the CPU does. I recently bought a new disk for my home Macintosh—mostly because it was cheap—and it just happens to have a particularly fast seek time. I have one copy of the system on it and a different copy on my older internal disk that came from Apple, and running from the new disk makes the system boot three or four times faster, and a lot of other things faster. You can't ignore I/O.

*KC:* No, absolutely. People have been asking me, when they go for new computers, "What's the most important component in your computer?" and I say "The [disk] controller."

*JS:* At Ridge we were very aware of that, and in the systems, we did all kinds of little things. We were one of the first to use SCSI's ability to manage overlapping seeks, because we often configured systems with multiple drives. We were devoted to exceptional performance in a number of ways and effectively, towards the end of Ridge's life, we found our strongest market niche which was "doing what you couldn't do with Suns." Apollo had vanished into HP by then, and Sun was really dominating the workstation market, but if you wanted to do what a Sun couldn't do, a Ridge was probably worth looking at.

*KC:* Give me an example.

*JS:* Sun boxes before SPARC couldn't have more than about 32 megs of main memory. Then in the first SPARC they had a problem with the design of the page table hardware, so that even when you had more than that installed, you had to switch memory partitions because they couldn't actually map more than 32 megs in one partition. The latest Ridges could have 256 megabytes of main memory—all addressable at once and with chips easily available at the time—which in 1988 was a lot even for a mainframe. That feature was one thing that sustained the company, because even though we weren't selling many systems, we made a lot of money off of the ones that had a huge amount of memory installed. Same for the ones with several disk drives. For people who were doing big CAD things, and using elaborate VLSI tools—VLSI was even bigger then than now, relatively speaking—it really mattered to do simulations. It was really important to have enough memory to get all of the state of your logic simulation co-resident in memory. Virtual memory really wasn't the same thing, even if you had plenty of swapfile.

*KC:* That's still true, and that's why Intergraph today is building boxes with a gigabyte of RAM.

*JS:* Intergraph's become a kind of niche company for connoisseurs somewhat in the way that Ridge was. We didn't set out to become a niche company, but we had designed in fewer limitations in some areas than happened to be designed into the competing systems, so that became a substantial influence towards the end of our life.

*KC:* Now, parenthetically, when someone bought what you might call a typical or characteristic Ridge system, what would it cost?

*JS:* I know at the end we got the minimum small-enclosure system down to $10,000, which was cheap in those days.

*KC:* Definitely.

*JS:* It was hard to say what "typical" meant, though. Yes, you could buy the bare bones for $10,000 and have a fairly versatile computer. But the customer who bought a Ridge because it was a Ridge was pretty much guaranteed to spend closer to $20,000, and if someone walked in wanting 128 or more megs in memory and the [disk] storage to match, they were on their way to a $100,000 system. Yet it was at that level, the very highest level, that we were most competitive—because so few other manufacturers could match our specification.

*KC:* Well, certainly in those days.

*JS:* In fact, another problem we had was that our least expensive systems didn't really find enough of a market. The low-end Ridge systems were as effective—in terms of both cost and computing power—relatively as the high-end ones, but a customer who wanted what we could deliver for $10,000 could pick from among more manufacturers, whereas one who wanted what we could deliver for $100,000 really had very few options. We were really most competitive with the big ones. We brought in enough revenue from the big ones to keep cash flowing, but if we had sold more of the smaller ones, we would have put our name in front of more people. The problem of course was, just as it is now, why do people buy PCs instead of Macs? Who's going to have the most apps ported to their system? Who's going to be here five years from now? What's the future?

*KC:* Let me ask a pointed question, then. Kunze seems to think that Ridge failed primarily because of lack of compatibility. But a lot of companies, including, I think, at this point Apple, have gotten into trouble by trying to build too many physical types of boxes, too many different configurations out the factory door. Was that a problem with you?

*JS:* No. We had two fundamental mechanical configurations—the small and the large box—and at any point in time they used the same circuit board. The customer would pick one box or the other depending on how much room they needed for I/O boards, for memory boards, for disks and for backup. You could just put more of everything in the big one. In the small box in particular, the CPU had to go in one slot, but I/O and memory boards could be intermixed, and in the large box there were enough slots that you could max out on memory and still have more room for I/O. So it was the best of both worlds—flexible enough that to the customer it was almost a custom system, and yet from the viewpoint of Ridge and Solectron every computer was, if you will, much more the same than different. We did have many problems—as you said, there's no single reason why Ridge failed—but the relationship between design and production was not one of them.

I might add, by the way, that Ridge didn't go bankrupt. We, the founders, saw that we wouldn't be able to survive in the long run without greater resources than we had been able to generate internally. The venture capitalists, by then, owned a majority of the company; Hambrecht and Quist had the leading role and Arthur Rock, who probably somewhat agreed with us, was in the background and less active at that point. We believed that the company should be sold to somebody, or merge with somebody, and the venture capitalists really didn't agree with that, and in particular didn't agree with the terms that could be achieved—they allowed us rather grudgingly to explore possibilities for merger or acquisition with various companies, but nothing came of that, and the VCs installed new management. It was as amicable as such a thing can be.

The biggest reason I would cite that Ridge was not more successful internally was that we, the founders, didn't get enough help from the venture capitalists. I don't mean money. I mean that they didn't take us by the scruff of the neck and knock sense into us, but they should have.

*KC:* They tried to replace your management a couple of times.

*JS:* Well, no, *we* did. We instigated those changes, including the departure of our co-founder and president, but they should have urged those on us long before we were ready. It was awkward for us because we were kicking out one of our own, but we finally did that, although we were embarrassed to say it that way. They probably weren't.

*KC:* Which one did you kick out?

*JS:* Dave Folger. But he was not the right person as president, and that was our failing. Neither were any of us. The rest of us only recognized that Dave didn't recognize that in himself. What we needed was the right sort of charismatic public face, as distinct from an engineer-president. And we all needed to think more strategically, more in terms of the big picture. Why was a company like Sun more successful? Sun got tremendous impetus by having Kodak, I believe it was Kodak, buy 25 percent of them; Kodak didn't do anything but sell the investment and take their profit, but Sun suddenly had a huge amount of money. At the time Kodak did that, Sun and Ridge were about the same size, but with all this cash Sun could hire people, they could afford to full-bore-design products and then abandon them, shelve experiments because they were too buggy or didn't give the yields. They didn't have to solve problems by being better engineers, they could solve a lot of problems with money. We meanwhile didn't have the smarts to see that some things other than engineering were important enough to do as best we could; we didn't understand that it might have been worthwhile to search for a charismatic leader who could schmooze with the CEOs of companies like Kodak and form the powerful alliances.

*KC:* Again, a lot of other companies had precisely that problem.

*JS:* So let's say that was our biggest mistake. I work for Hugh Martin at 3DO now and I think he's turned into that at this point. I mean, I'm 3DO's chief technical officer and I continue, frankly, in a lot of the roles that I had even in the Ridge days, but Hugh has evolved from being the engineer he was at Ridge to being someone capable of doing all that it means to lead and run a large business, and work with other large businesses. If we were building Ridge over again today, even with the same personnel, I think the outcome might be a lot different. Our other mistakes, like the UNIX issue, didn't help matters of course, but I think the lack of a corporate attitude, if you will, was fundamental.

*KC:* So what do you think, in the last analysis, Ridge gave to the history of computing?

*JS:* We did spur a lot of other people to develop architectures that were similar to ours and to use techniques that we pioneered, whether or not the use is attributed to us. One [company] that does directly attribute in some ways is HP. I have a good friend, a senior person at HP, who's told me that in 1982, when HP hired Joel Birnbaum and Bruce Worley, those two had to put up a fight within HP to start what became the Precision architecture, right about the time when the first technical articles came out about what was inside a Ridge system, and I'm told that they were brandishing some of the earliest articles about Ridge and its architecture around HP, trying to say, look, this is what we should do. So even if the Intel/ Microsoft duopoly swallows up everything, a lot of architectures including HP's have probably been influenced directly or indirectly by what we did. Certainly a lot of the techniques that we implemented, in terms of out-of-order execution and cache access and so forth, you're starting to see in Pentium chips. Our method of static branch prediction has been added to both Power PC and MIPS. A lot of people have gotten patents since on things for which Ridge accomplished prior art, but since the company's gone, no one's there to care.

*KC:* And that's very difficult. But it has to be said ultimately that especially in the context of the components being used, Ridge was very, very advanced technically and theoretically.

*JS:* I think it was, both of those, but we all know that the best technology doesn't win simply on its own merits. Now, it doesn't *hurt.*

*KC:* I happen to think that if the Wintel duopoly tries to stretch to cover everything, some holes are going to appear in it.

*JS:* Oh, yeah. I don't mean that they're going to succeed forever, but they certainly are living proof that they don't have to be the best. Nice if they do, but not absolutely necessary, for Microsoft, even more than for Intel. You'll be more successful cloning Intel, as there are people doing, if Intel screws up than you probably will trying to take on Microsoft when Microsoft does something less than great; although if Microsoft does something terrible, they won't succeed even given who they are. But Microsoft just has to make [Windows] NT be good, technically okay, and it'll be successful.

*KC:* Right, and NT is really okay.

*JS:* It is. It could be called pretty good, so it'll succeed, in my opinion.

## IN MEMORIAM: CUTHBERT C. HURD

**by Dag Spicer**

Cuthbert C. Hurd, one of IBM's top visionaries in the early 1950s and a prime mover behind the famous IBM 701, died at his home in Portola Valley CA on May 22, 1996. He was eighty-five.

Hurd was born on April 5, 1911, into a family of farmers and itinerant preachers in Estherville, Iowa. He showed early promise in mathematics and earned a bachelor's degree in that subject from Drake University, a master's from Iowa State College and a doctorate from the University of Illinois. He then embarked on an academic career of some distinction, and during the war years served in the U. S. Coast Guard Reserve, before he was asked to set up a computing service at the Oak Ridge National Laboratory. There he conceived and directed the deployment of IBM punch card machines to the solution of classified research problems in the nascent U.S. atomic energy program.

On January 27, 1948, Hurd first met Thomas J. Watson sr. and jr. at the dedication of IBM's Selective Sequence Electronic Calculator (SSEC), the fastest calculator in the world at that time and the successor to Howard Aiken's Harvard Mark I, a machine largely financed and developed by IBM but for which the credit was deliberately obscured by Aiken. The SSEC was IBM's rebuttal to Aiken's pride of place and, at the same time, a declaration of substantial intent in the world of scientific and commercial computing. Thomas J. Watson sr. was impressed by Hurd's combination of practical and theoretical talents, and lured him away from his work at Oak Ridge. Hurd was only the second Ph.D. hired by Big Blue—Wallace Eckert was the first.

Though Hurd was considered one of the country's top three mathematicians in the 1940s, he was naturally gregarious and at ease with people from all walks of life. He was ideally suited to confer with the thousands of scientists and engineers who sought new computational tools which would solve the myriad problems arising from new disciplines fostered by WWII. The advent of the Korean War brought still more acute need for high-speed calculation in these fields.

Hurd found himself confronted with a challenging assignment. A younger generation of engineers, familiar with wartime advances in electronics, keenly felt an intuitive but indistinct premonition that a single machine could be flexible enough to address problems of general computation. Against this evanescent optimism was posed a legacy of huge punch-card hardware profits and a discouraging history of losses on research machines. Calculators had been cast in a mold of immense and narrowly dedicated "giant brains," gargantuan devices like Mark I, SSEC, and ENIAC, which could take weeks of setup time for problems which they would then run in hours or even minutes.

The situation was confused, but Eisenhower's declaration of war in June of 1950 left no room for indecision. Hurd embarked on an exhaustive cross-country tour of military and aviation installations, mapping the contours of the machine which would address the most computationally demanding problems. In spite of his august patronage by the Watsons, he faced a protracted battle within IBM as more conservative elements, doubting the need for a radically new machine and citing IBM's handsome profits in punch card equipment, sought to curtail his efforts.

But the ascendancy of a new class of machine was foreshadowed by the introduction, in 1949, and tremendous success of the Card Programmed Calculator (CPC). This was IBM's refinement of a hybrid, easily modifiable stored-program machine invented by two engineers at Northrop Aircraft (a major IBM customer) who, working with Northrop's own superb machinists and tools, had surreptitiously ganged together a Type 604 Electronic Calculating Punch and a Tabulator. IBM executives, angered by Northrop's modification of leased machinery and embarrassed at having a customer innovate so starkly, responded with sputtering impatience. Hurd by contrast saw an opportunity, took over the project, formalized the invention into the CPC and negotiated rental agreements of up to $5,000 a month. Such immense profits forced even IBM's old guard to look more closely at the possibilities of electronics.

The question of a calculating machine considerably improved over the CPC arose quickly. At week-long meetings over the Christmas holidays of 1949, IBM executive staff debated the pros and cons of development and finance, but came to impasse after impasse. Finally, Tom Watson jr. broke the deadlock and said "Hurd, if you can find the customers, you can have the money." Watson also specifically admonished those present who had not supported Hurd's program to "stay out of his way."

Right after the meeting, which Hurd later described as a "shootout at OK Corral," he and IBM chief engineer Nat Rochester connected the block diagram describing the Institute of Advanced Study machine (from a report by von Neumann, Burks, and Goldstine) to representations of a magnetic drum, magnetic tapes, and punch card I/O. Armed with thirty copies of this rudimentary diagram, Hurd and a sales manager from headquarters took the train out west and spent the next three weeks jawboning IBM's largest customers. Despite an anticipated rental of $8,000 a month, Hurd and his colleague Jim Birkenstock—who canvassed east coast customers—collected about 25 letters of intent. IBM management was convinced to commit resources to the new project.

The proposed machine was provisionally called the "Defense Calculator" to garner internal IBM support by appealing to Watson sr.'s patriotic bent, while paying lip service to the company line that IBM did not build "computers." Expensive but powerful, it would contain 4,000 vacuum tubes at a time when the largest research machines had scarcely more than 1,000, be completely modular for efficient setup at site, and be finished and documented to IBM's usual imposing standard.

Within two years—on April 7, 1953—the new machine, now officially christened the IBM Type 701, was officially introduced to the world at IBM's New York headquarters before a glittering crowd of the nation's scientific and military elite. The keynote address, by "father of the atomic bomb" Robert Oppenheimer, stressed the usefulness of the new "Electronic Data Processing

Machine" in such applications as weather prediction, astrophysics, and atomic weapon design—problems which "...were deeply non-linear [and which] went beyond the familiar mathematics of the last centuries." Hurd then officially demonstrated the machine to this inaugural audience by solving a neutron-scattering problem, which typically took humans six weeks to complete, in a mere six minutes. The 701 proved to be a sought-after machine, and although monthly rental ballooned from the promised $8,000 to an eventual $18,500, nineteen such computers were built and installed from coast to coast. The success of the IBM 700 series was Hurd's success as well, and in 1953 he became the company's Director of Applied Science, then in 1955 was named Director of Electronic Data Processing Machines.

The 701 was followed by machines such as the 704 and 709 which featured improved performance, memory size and diversity of instruction set (the 701 had 32 instructions). Hurd supervised the entire program and also served as IBM's "front man" for media relations, appearing on dozens of radio and television programs describing IBM's machines to an often incredulous lay audience.

Yet Hurd's public position and duties tell only part of the story. He traveled tirelessly behind the scenes, among the nation's top scientists, seeking to understand the form and style of computational problems and solutions. He discussed problem-solving with Edward Teller, Enrico Fermi, Albert Einstein (who told Hurd the only computer he needed was a pencil), and Werner von Braun. His greatest coup was undoubtedly enticing his close friend John von Neumann to become a consultant for IBM in late 1949, an association which greatly influenced the design of the 701 and subsequent machines, and effectively quelled any opposition within IBM to the direction Hurd was taking the company.

Though a consummate "IBM man," Hurd greatly admired ENIAC designers Presper Eckert and John Mauchly for their courage in developing the UNIVAC brand of commercial computers at a time when the market for such machines was very dubious. IBM could sell primarily to defense contractors, universities and insurance companies—entities still very accustomed to calculating with punch cards—whereas the more ambitious UNIVAC system employed not only punch cards, but magnetic tape, a data medium which Watson sr. passionately distrusted. Hurd took the competitive challenge to heart and lobbied hard for a comparable IBM entry. The ultimate result was the IBM 650, a wildly popular machine which became IBM's first mass-produced computer with some 4,000 sold. The machines were either given outright or leased at attractive rates to universities, and defined a canonical architecture for a decade or more, earning profits for IBM which were unsurpassed until the introduction of the System/360.

One key to Hurd's success was his ability as a listener. A second was his perception of the value of customer education. At a time when the title "programmer" simply did not exist, Hurd began a series of customer education classes and computation forums which brought together scientists and people from industry to share solutions to common problems. Even companies that competed brutally for market share met at Hurd's sessions to exchange algorithms and source code. These meetings eventually coalesced into informal user groups of which Los Angeles-based SHARE, founded in 1955, was perhaps the best-known and best attended. Under Hurd's guidance, John Backus created FORTRAN; other "Hurd men" created assemblers and all manner of utilities, and acted as troubleshooters for application-specific software development.

From 1956 to 1961, Hurd oversaw the STRETCH project which produced the Model 7030, IBM's first transistorized supercomputer, widely acknowledged as the first machine to embody many concepts later implemented in the 360 architecture.

In 1961 Hurd left IBM as the result of an internal political squabble—in which he was wholly innocent—but remained a consultant to the company until 1985, in the meantime forming Computer Usage Company and publishing two influential books on the System/360. He brought CUC to $15,000,000 in sales in its first five years by providing programming and systems support to System/360 adopters. Hurd left CUC in 1974 and over the next twenty years was the motive force behind several computer start-up companies, the most successful of which, Quintus Corporation, was acquired by Intergraph in 1989.

Hurd's contributions to computer science were pivotal. He was one of very few individuals of his day who could translate the mathematical formalisms of contemporary scientific problems into the parallel terminologies of machine architecture and software. He was ideally qualified to serve as an intellectual bridge between mathematicians and engineers on the one hand, and IBM's management and technical staff on the other.

Hurd passed away peacefully on May 22 of last year, surrounded by his family at his home. His life was unique in its confluence of outstanding technical merit, historical opportunity, and estimable personal qualities. Always loyal to his modest roots, he nonetheless went far beyond them. The ANALYTICAL ENGINE extends heartfelt condolence to family, friends and colleagues.

*Dag Spicer, collections manager of The Computer Museum History Center, is at work on a biography of Cuthbert Hurd.*

## IN MEMORIAM: SEYMOUR CRAY

**by Erich W. Schienke**

Seymour Cray, "father of the supercomputer" and one of the industry's most brilliant and idiosyncratic designers, passed away at Penrose Hospital in Colorado Springs CO on October 5, 1996. He died a week after he sustained severe head injuries when his vehicle was involved in a multi-car collision on a nearby Interstate highway. He was seventy-one.

Born in Chippewa Falls WI on September 28, 1925, Cray was an archetypally straightforward Midwesterner who set out "to do what had to be done." His dislike of formality, distraction and paperwork was so deeply rooted that it earned him a reputation for eccentricity which, while sometimes exaggerated, was hardly undeserved. Many of life's daily pursuits and rituals were, for Cray, simply irrelevant to the central and honest implementation of one overwhelming desire—to make a computer go as fast as possible, again, and again, and again.

Cray already showed considerable talent in radio and electrical design and construction by the time he graduated from high school in 1943. He served in the U. S. Army infantry in Europe during the last months of World War II, then briefly in the Philippines, and returned to the University of Minnesota, from which he graduated with a bachelor's degree in EE and a master's in mathematics in 1951. Casting about for the logical next step in his career, he asked a professor for advice and was told to go "down the street to the glider factory"—Engineering Research Associates, then engaged in the design of the famous 1103.

ERA gave Cray an excellent start as a computer designer and left him with many warmly remembered experiences, such as meeting John von Neumann. It gave rise to the philosophy of simplicity in design which he later summed up as "Don't put anything in that isn't necessary," and which eventually earned him credit as the originator of RISC. But as ERA merged with Remington Rand and then became Sperry Rand, accumulating layers of management as naturally and silently as falling snow, Cray began to feel his equally legendary impatience with bureaucracy and preference for small, purposeful work teams. In 1957, at the invitation of Bill Norris, he left ERA to become a co-founder of the leaner and meaner Control Data Corporation. (ERA, after many more years and several more mergers, ended up as a sliver of Unisys.)

Control Data was the first computer company that built only with semiconductors, with no prior tradition of vacuum-tube construction. Cray's initial project at CDC was the 1604, a large machine which was not the first commercially available transistorized computer (it was preceded by machines from Univac and RCA) but was certainly the fastest. Under development by 1958, the 1604 was first delivered to the U. S. Navy on January 10, 1960; it brought attention to CDC and Cray—especially from IBM, which accelerated the STRETCH project as a response. Other CDC computers kept up the tradition of simplicity and speed, and the Model 6600, announced in August 1963 and delivered to Lawrence Livermore National Laboratory about a year later, is now generally considered the first "true" supercomputer. The 6600, delivering 3 MIPS, was stunningly quicker than IBM's contemporary top box, the 7094; Tom Watson jr., not letting CDC out of his crosshairs, reported in a scathing internal memo that the company employed thirty-four people "including the janitor" and pointedly asked how an engineering staff of that size could ever have produced the world's fastest computer. Unrelenting rivalry between IBM and CDC later produced such oddities as the shadowy System 360/91 and the earth-shaking, five-year "straw computer" lawsuit.

But the success of the 6600 and its graceful successor, the 7600, was too much for Seymour Cray; it made him into a semi-public figure and CDC into a big, cash-rich company full of managers. He had already convinced Norris to move Control Data's R&D lab from Minneapolis to Chippewa Falls, and in 1972—after the cancellation of the CDC 8600 for what he considered insufficient reason—he broke with CDC to start Cray Research (CR). The CRAY-1, introduced in 1976 just after CR went public, was Seymour's and CR's first parallel-processing machine built with integrated circuits; it was also only the second computer to be built as a vector processor, achieving its speed by chaining calculations, rather than by splitting them as in "ordinary" parallel multiprocessing. In a startling expression of Cray's dictum, "The best connection between two components is the shortest possible line," the CRAY-1 had a backplane bent into a C-shape and surrounded by a bench-like casing for the cooling hardware that gave it the nickname of "loveseat computer" to this day.

Cray proposed another great leap in systems design—and nearly a disastrous one—with the CRAY-2, designed as a smaller, faster and more powerful version of the CRAY-1, but with gallium arsenide (GaAs) processors and the famous fluorocarbon immersion cooling system for circuit boards. The GaAs processors were beyond the fabrication techniques of the day—and today; eventually Cray had to redesign the CRAY-2 to accept silicon circuitry, and by the time he did, it was far behind schedule, finally appearing in 1985.

In the interim, Cray Research survived only because the company hired a second lead designer, Steve Chen, to create an upgrade of the CRAY-1 which became the successful CRAY X-MP. Cray and Chen could not work in harmony. The experimental CRAY-3, again using GaAs processors, consumed vast amounts of money and seemed destined for the same failure as the early design of its predecessor. In 1989, Cray moved to Colorado Springs and founded Cray Computer Corporation (CCC) to continue development of the CRAY-3, which unfortunately cost $300 million by the time its design collapsed. Cray attempted to regroup with the CRAY-4, a silicon-based machine with 64 processors that pushed the boundaries of connectivity. This computer was incomplete when CCC filed for bankruptcy in 1995, yet Cray had no intention of backing away from risks. He was still at work on enhancements at the time of his death.

Seymour Cray, who knew every connection in his machines and expressed almost a Bauhaus esthetic in their implementation, will be remembered as one of the industry's most innovative and uncompromising creators. His accidental death is doubly sad because it seems to lower a final curtain on the era of the cutting-edge, cost-no-object supercomputer designed with genius and built by hand. His rules of thumb, such as "Make the bandwidth to memory as broad as possible" and "Don't couple a fast processor to slow I/O," will remain an inspiration to younger engineers, even those who work on the microprocessors that hastened the supercomputer's decline. The ANALYTICAL ENGINE offers its condolence to his wife, Geri Harrand Cray; his daughters, Susan Cray Borman and Carolyn Cray Arnold; his son Steven Cray, and his colleagues and friends throughout the world.

*Erich W. Schienke is the general manager of the Computer Institute and assistant editor of the ANALYTICAL ENGINE.*

## DID YOU HEAR ANYONE SAY GOODBYE?

**by Donald A. Thomas, Jr.**

It's really amazing to me that an institution as big and as powerful as Atari—or as Atari once was—can have been shut down in recent days without a perceptible flinch from within or outside the gaming industry. I can understand that gamers wanted to push Pong out the door early in the timeline. I can appreciate that classics such as Missile Command and Asteroids do not push the technical envelopes of 32-bit and 64-bit systems. I know all these things intellectually, but the heart cannot face the final truth: The world and the corporate machine known as Atari could not find an amicable way to coexist.

On Tuesday, July 30, 1996, Atari Corporation took each and every share of its company, ATC, wrapped them all in a tight bundle and presented them to JTS Corporation, a maker and distributor of hard disk drives. On Wednesday, the shares were traded under the symbol of JTS. Most of the staff of Atari was dismissed or resigned, and within a few weeks, the remainder—three people —moved to JTS' headquarters in San Jose, California. The ex-Atari employees were assigned to different areas of the building. All that really remains of the once-great Atari name is a Santa Clara warehouse full of unsold Jaguar and Lynx products. [*Note*: On December 23, 1996, JTS unloaded all remaining Jaguars to a national catalog closeout company, Tiger Software. As of January, that company offers Jaguar systems plus four titles for $59.99.]

As recently as the middle of 1995, Atari executives and staff believed that things were finally taking a better turn. Wal*Mart had agreed to place Jaguar game systems in 400 of their Superstores across the country. Invigorated by this new hope and the promise of opportunities that open when such deals are made, Atari invested heavily in the product and mechanisms required to serve the Wal*Mart chain. But the Atari decision-makers clung to their favorite philosophical belief—that great products never need advertising or promotions—and put the Wal*Mart deal straight into a tailspin. With money tied up in product on shelves and paying the costs to deliver them there, not much was left to saturate any marketplace with advertising. Most game-savvy parents rushed into stores to get their kids Saturns or PlayStations; the few who picked up the Jaguar were chastised by disappointed children on Christmas Day.

In an effort to salvage the foundering Wal*Mart situation, Atari instigated desperate attempts to run infomercials across the country. The programs were professionally produced by experts in the industry and tailored to permit Atari to run slightly different offers in different markets. Although these infomercials were relatively cheap to run, they were very costly to produce and support, and the results were disappointing. Of the few thousand people who actually placed orders, many returned their purchases after the holidays. The kids wanted what they saw on TV during the day! They wanted what their friends had! They wanted what the magazines were raving about!

In early 1996, Wal*Mart began returning all remaining inventory of Jaguar products. After reversing an "advertising allowance" Atari was obligated to accept, Wal*Mart left Atari with the net benefit of an overflowing warehouse of inventory in semi-crushed boxes and with firmly affixed price and security tags. Unable to find a retailer willing to help distribute the volume required to stay afloat, Atari virtually discontinued operations and traded its remaining cash to JTS in exchange for a graceful exit through the industry's back door.

It's worth remembering that Atari Corporation — not Nintendo—actually introduced the first 64-bit game system on the market, just before Christmas 1993. Since Atari couldn't afford to launch the Jaguar nationwide, the system was introduced in the New York and San Francisco markets first. Beating the 32-bit Saturn and PlayStation game decks to the punch, Atari enjoyed moderate success with the Jaguar and managed to lure shallow promises of development from third-party companies. Unfortunately, programmers grossly underestimated the time required to develop 64-bit games; the jump from 8-bit and 16-bit was far wider than anticipated. In addition, Atari was already spread thin monetarily, but was prevailed upon to finance almost every title in development.

An assortment of games began to hit store shelves almost a year after the initial launch, but by then the '94 holiday season was a lost opportunity. Even then, Atari de-emphasized many of the planned titles to minimize problems caused by hasty coding and QC. Consumers were unhappy and retailers were dismayed. The few ads that Atari was able to place in magazines often ended up giving incorrect release dates because magazines, deadlined up to 120 days in advance, fell behind Atari factory information that changed almost every day.

The demise of the Jaguar, clearly, would be an exemplary comedy of errors, if only it had been funny to the last standing group of dedicated consumers. Was this the first time Atari tried to launch quality product without adequate marketing resources? Unfortunately, no; it was only the last of a long series. For some of the rest of the story, travel with me back to 1983....

In that year Warner Communications handed Jack Tramiel the reins of the company. By this time, Atari was often considered a household name, but few households wanted to spend much money on new software, and the systems—the VCS 2600's, 400's, and 800's—were lasting forever. Too few elected to buy new ones. Atari's own insistence on product quality, played off against Warner's obscene spending, resulted in a *daily* loss of over $2 million. Atari was physically spread all over the Silicon Valley with personnel and equipment in literally 80 separate buildings, not counting international offices and manufacturing facilities. This was a company whose own lack of direction went to its very roots.

Tramiel took only the consumer branch of Atari, forced Warner to deal with the arcade division separately, and implemented his "business is war" policies. Within a few years, he took the company public, introduced an innovative new line of affordable 16-bit computers—the beloved ST series—and released the 7800 video game system. Many hailed him as a miracle worker although he never felt like one. Gratitude for success was something Jack never understood; success comes from hard work, and hard work was the level of performance he expected.

People who publicly quoted his statement that "business is war" often felt that it implied extreme aggressiveness in the marketplace. The meaning actually had closer ties to Tramiel's experience as a concentration camp survivor—experience suggesting that no extreme measures in business could equate to what he knew people could be forced to face and cope with in life. Of the 80 buildings in Sunnyvale, Santa Clara and Milpitas, almost every one was amputated from Atari's body of liabilities. The people, the work, the heritage, the history were fired or liquidated. Those who survived were unsympathetically required to fill gaps and, while most tried, few found a way to successfully do the work of a dozen predecessors. Atop the mountain, Jack pressed with an iron thumb. All Fed/Ex mailings were required to be pre-approved by one of a handful of people. "Unsigned" purchase orders went unpaid regardless of the urgencies that inspired their creation. Employees found that they could only accomplish their tasks by spending valuable time trying to find ways around the system; some never found a way to make things work, and lost their jobs, while others bent the rules and lost their jobs just as quickly. But as horrible as this all sounds, it actually was the only way to protect Atari as a company and give it a chance to survive—as it did, for a time, very well.

Jack's 16-bit computer, the "Jackintosh" or Atari ST, received a hearty welcome in both the United States and Europe, but would at last have a far greater impact in Europe and especially Germany. Europeans were delighted with powerful "affordable" technology and relatively unconcerned with the IBM compatibility of which Americans made a fetish. Jack Tramiel was delighted, too, at explosive growth of market share in Europe where retail prices were much higher than Americans were willing to pay. As a result, most of the ST production was being shipped to European destinations to capture the higher margin. This enraged Atari loyalists in the United States, who were waiting months for local stores to take delivery, while they read international magazines touting ample supplies. Those in the U.S. who followed Atari closely became dismayed at the company's slow, seemingly heedless abandonment of its name recognition. More casual Atari users merely wondered at the lack of developments, and some simply assumed that the company had filed for bankruptcy.

Technically, Atari 16-bit computers were designed beyond their time. For less than $1,000, consumers could enjoy "multimedia" before the term was ever widely used. The windowing and iconic interface—based on GEM from Digital Research, the developers of the pioneering micro operating system CP/M—provided a stable, flexible and highly functional environment long before the heyday of MS-Windows, with which it shared many essentials. The ST's built-in MIDI became an instant hit in the high-end music industry. Tasks were activated and manipulated with a mouse, and the system had an unusual variety of ports for industry-standard peripherals such as printers and modems. Diskettes were the then-new Sony stiff-shell 3.5-inch form factor.

The genius that went into the technology of the machines stopped well short of inspiring their promotion and marketing. Mr. Tramiel, the founder of Commodore Business Machines, discovered that when he introduced the PET computer in 1977, he didn't have to call a single publication. Reporters from news magazines, science journals, business publications and newspapers flocked to his door demanding an opportunity to see the product. They arrived with microphone, camera and pen in hand, and they kept coming back. An added switch, a new 4K application, or the signing of a new retailer were all big stories the press wanted to handle. Jack's early experience with the romance of commodity computing taught him that the world would beat a path to the door of a company making a better mousetrap.

Clearly, by Atari's day this no longer applied. The world had revolved a few times beneath Tramiel and he never noticed. The tactics successfully used to sell Commodore computers had become antiquated notions. Today, a new video game announcement may generate a request from any of the dozens of gaming magazines for a press release, but a lot of costly work has to be done to assure fair or better coverage. Editorial people are swamped with technical news and bombarded with samples. Faxes fly in through the phone lines and e-mail jams up their hard drives. It takes a lot to grab their attention.

Atari's marketing people hoped for success with the Jaguar, on its considerable merits, but they were fighting established standards in the industry with severe handicaps imposed by policy. The Jaguar was and is primarily a cartridge-based system; since cartridges were so expensive, editorial people were required to return them before new ones would be sent. Editorial staff like to assign review projects, and could rarely recover cartridges that they had sent out to their stringers. Meanwhile, reviewers—who often love their work because they get to keep what they write about—felt insulted by Atari's "business is war" penny-pinching. This did leave a few magazines willing to cover Atari products, but their requests were often turned away because of a shortage of programmable cartridges, indecision by marketing, or any number of other vague barriers. In-store signs and posters were occasionally produced, but many retail chains charge premiums to manufacturers that want to display them. Some direct mail campaigns were implemented, but Atari's erratic development schedule often meant that advertisements were published and distributed for products that were not yet available.

Meanwhile, the PlayStation was launched with over $500 million in marketing funds—and even that amount only bought Sony a fierce rivalry for first place with the Nintendo 64. One of these platforms, almost certainly, will emerge as the most successful next-generation gaming machine throughout the world. Tramiel, the great survivor, never learned how to compete on that scale and, in any case, could never have afforded it.

As the 1990's got underway, a paradigm shift caused crucial damage to Atari's computer division. Europe and the rest of the world discovered that IBM-compatible computers were becoming more powerful and more affordable. Volume production held out the promise of the same applications, the same interface, and the same communications at home as at the office; companies like Dell and Gateway made fortunes by turning a trend into a market as the increasing power of IBM-compatibles migrated from office to home, then brought the office home with it. [This may not have been an unmixed blessing—Ed.] Companies like Commodore, Atari and NeXT couldn't compete any longer. The dedicated user base of each platform felt abandoned as its company was forced out of the computer market, but the inevitable prevailed; Commodore jumped ship, NeXT changed business goals completely, and Atari invested what they had left in the Jaguar game system. Today, even Apple is kicking and screaming—having created a huge niche for themselves by focusing heavily on education, only to discover that when kids grow up and get jobs, they want computers that follow the prevailing business standard....which is too often not the Macintosh.

Sadly, this has meant that the history and talent of Atari has been too little appreciated over the years. Few people other than collectors and historians remember Atari's truly trailblazing hardware like the TT030 and Falcon, the STacy, the Atari Book, and the Portfolio. Atari's founder, Mr. Nolan Bushnell, went on to create Chuck E. Cheese Pizza. Apple Computer was started in a Los Altos garage by ex-Atari employees. Activision was founded by Atari's most formidable game programmers. Hewlett-Packard followed the form factor of the Portfolio palmtop when it introduced the consistently successful LX series. Atari, like Fairchild, has become a company better remembered for its imprint than for itself.

But for some pathetic reason Atari's final days came and went with no tribute, no fanfare and no dignified farewells. Why? Where did all the talent go? Where are the archives? Where are the vaults? Where are the unpublished games and where are the master copies of the published games? Why has no company stepped forward to adopt Atari's remaining intellectual property? Where are the creditors? What has happened to the properties and sites? Where are the databases, warranty cards, promotional items, notes on meetings, unanswered mail? Who owns P.O. Box 61657? Who goes to work in Atari's old offices? Where do consumers have their systems fixed? Who is publishing new games? Who still sells Atari products? Why are a lot of people still talking about Atari on-line? To me, these questions all deserve an answer, but who will ask them?

You can and I wish you would. If you believe Atari left us without a word, contact Dateline at dateline@nbc.com. Send this article to ten of your friends in e-mail. Mail copies to newspapers or other news programs. A letter in your own words would be great!

I'm an ex-Atari employee and proud to have been. I'm still an Atari devotee and proud to be. I'd spend money for a thorough retrospective on Atari. Wouldn't you? Wouldn't it at least be nice to say goodbye?

*Don Thomas can be reached at 75300.1267@compuserve.com.*

## BAGGING A JAGUAR

You can have your very own Atari Jaguar 64-bit game deck, an instant collectible, for no more than a phone call and a discreet notch in your credit card. [Shameless plug!]

On December 23, 1996, JTS unloaded all remaining Jaguars to a national catalog closeout company, Tiger Software. As of our press date, that company offers Jaguar systems plus four titles—Raiden, NBA, Alien Predator and Cybermorph—for $59.99. The toll-free number is +1 800-888-4437. This is an overstock liquidation and, naturally, neither the vendor nor the CHAC can guarantee availability—but we'd recommend you take a shot. We did!

## Book Review: WHERE WIZARDS STAY UP LATE Katie Hafner and Matthew Lyon

**Reviewed by Erich W. Schienke**

In these bizarre and hyperactive days, when so many people seem to think that the Internet and everything attached through it sprang full-blown out of nowhere, its real history takes on a special importance. In fact, the story of the Internet—and its predecessor, ARPANet—can be any one of several stories that share only facts, references and technical details involved in the development of the network. Katie Hafner and Matthew Lyon choose to tell the "roots" story of the Internet by concentrating on its origins and originators.

The book opens with a consideration of computing and scientific research managed by the Pentagon at the time of the Cold War. We quickly become familiar with the Advanced Research Projects Agency (ARPA) and its high-risk, high-payoff approach to military-funded science, a product of the post-Sputnik frenzy. Introducing us to the psychoacoustician J.C.R. Licklider, first director of ARPA's Information Processing Techniques Office (IPTO), Hafner and Lyon describe the first, brilliant, theoretical conceptions of computers in terms of a network and shared resources—this at a time when few had thought even of making two computers communicate with each other. The enthusiasm and vision of Licklider and Bob Taylor (Licklider's replacement and mentor in the IPTO,) are wonderful to watch as the ARPA networking project is born and endowed with its first million dollars of research support.

Wizards then shifts its focus to the efforts of RAND employee Paul Baran who, with exemplary back-of-the-envelope calculations, brainstormed the relative advantages of centralized systems and distributed networks, concluding that the best way to exchange messages between computers was to break each message down into multiple packets and reassemble them upon receipt at the host machine. Packet switching, we discover, was a concept developed independently within the same year by Baran at RAND and Donald Davies in England; but this very robust technology faced an uphill struggle because it was seen as ludicrous by the likes of communications colossus AT&T. Unfortunately, we never really learn why they thought it was a ridiculous idea, nor do we find out as much about Donald Davies as we do about Paul Baran.

Larry Roberts, heading up the implementation of the ARPA network project, put the building of the network—and, in particular, the development of the "interface message processors" ever afterward known as IMP's—up for bid. After some description of the other major companies' bids we learn that Bolt, Beranek & Newman (BBN Corporation) had the most evolved ideas and best ability to implement them. In retrospect, even allowing for this book's highly pro-BBN perspective, they seem to have been a shoo-in for the contract; it made sense that the management of packet switching should best be run by a small, highly efficient company, rather than one bogged down with layers of bureaucratic difficulties, like AT&T, Honeywell, or Raytheon. If that was one of Roberts' motivations in choosing BBN though, it was only one; BBN seemed the obvious choice for many reasons, not least because they had the likes of Frank Heart (one of the seldom sung, if not unsung, heroes of the ARPAnet) and Bob Kahn.

Deciding on the Honeywell 560 as the first IMP (later supplanted by the 316), a rugged machine intended to keep pesky grad students in check, BBN went forward with heavy debugging and building up the first four nodes of the ARPAnet by the beginning of 1970. Over the next four years the system grew relatively quickly, but the traffic was not growing in proportion.

As the mid-70's became the mid-80's and ARPANet transmuted into the Internet, major changes took place in the management and use of the system. The last third of Wizards describes how the centralized control of a distributed network, ARPA, inevitably became decentralized in control—of how the IMPs were maintained, and of what content was sent over the network. Hafner and Lyon are great at giving us glimpses of early free speech debates, the emergence of netiquette, and the enthusiasm with which people took to communicating over the net. People, clearly, were as much a determinant of the net's character as were computers.

We learn how Vint Cerf and Bob Kahn took the ARPANet (by then DARPA) global by connecting it to the European CATENET, abruptly adding radio and satellite links to the secure standard of uploading and downloading via wires. The IMP protocols became obsolete in a hot minute as the Wizards buckled down to development of a communications protocol that would allow the ARPANet to communicate with the unreliable CATENET and whatever networking solutions lurked unseen over the horizon. Cerf and Kahn published their Transmission Control Protocol/Internet Protocol (TCP/IP), which established a worldwide standard for communicating between networks, spawned the Internet as we know it, and is still a thoroughly meritorious piece of code.

The book glides to a close with a brief introduction to Bob Metcalfe and his conceptual development of smaller local area networks (LANs) through the innovation of Ethernet, invented as a companion to the Alto at Xerox PARC. Crucial to high-speed transfer of data on local machines, Ethernet also transformed the sharing of machine resources within the local context of a campus or building, eventually becoming a commodity and greatly improving flexibility of 80's and 90's microcomputers.

If that's what's in this book, what isn't? First of all, the ARPANet was very much a "net" of physical cabling, yet Hafner and Lyon take the "wires" almost for granted. AT&T, they concede, kept up its part of the contract with ARPA without a problem, delivering 50-kilobit lines where and when they were needed. But what went into a line noise-free enough to carry 50Kb in 1969? What wizardry was implicit in transferring digital packets and sending them over a line via an analog modem? I would have appreciated learning more about those topics in reference to the 'origins' of the early networks. More detail about communications research in universities would have helped me understand the balance among educational, military and private sector participation in building the

ARPANet. I also craved a better overview of what development was done regarding sharing work and resources over the network. And oh, yes—what ideas *didn't* work in implementing the net? As history, this book has its merits; as a history of how the net became what it was, as opposed to something else, it leaves too much as obvious and occasionally becomes diffuse.

Still, let's not judge a good book by what it's not. An informative survey of the technologies involved in early incarnations of the net, this book really shines as a history of the Net's great thinkers and builders. In Wizards, character counts for as much as accomplishment when individuals are introduced. Hafner and Lyon do a great job of making the story human, of expressing the concerns and delineating the lifestyles of techies and hackers, and of describing the Internet as a culture that grew along with its technology—a poignant consideration now that the tables have turned and the Net is having an impact of its own on popular culture. Wizards deserves a careful read and can teach much, if not everything, about what we now know as "our" Internet. The serious student of computers and communication will need to go beyond this book to learn the history of the Net—but definitely won't skip it, either.

## SPOTTER FLASH

The last quarter of 1996 has been a busy time for the CHAC and its colleagues, who reveled in a flood of media attention—including our first television feature! But, magazine and television lead times being what they are, all of this appeared or will appear in the first quarter of 1997—so we'll tell you all about it in ENGINE 4.1.

## THE COMPUTER MUSEUM HISTORY CENTER

### Len Shustek

The roots of the electronic computer reach back only 50 years, yet the Information Revolution that it has produced is becoming as important to civilization as the Industrial Revolution was. What are we doing to preserve the record of how it happened?

Only lately has there been a general awakening of interest in computer history, fueled by the sudden awareness that even recent history not preserved is history lost. Everyone now recognizes that two crucial keys to that history are being lost at an accelerating rate: old computers, and the people who created them. We owe it to ourselves and our descendants to preserve, study, and celebrate the history of this incredible invention and its impact. If we don't, it will be lost.

The Computer Museum, based in Boston, is a successful organization that has long been dedicated to the preservation of the history of computer technology. TCM is now undergoing a major expansion to the west coast and establishing the Computer History Center in Silicon Valley.

Why the Valley? The computer was not invented in the San Francisco Bay area, nor are most of the world's computers built there. But few would argue with Silicon Valley's current claim to the title of intellectual capital of the computer business. The confluence of established companies, startups, venture capital firms, technology-friendly universities, and media coverage has caused the center of gravity to shift to this area. That geographical imperative, combined with the resurgence of interest in preservation, makes it clear that now is the time, and Silicon Valley is the place, to establish a new major computer history center.

The Center has been created as the west coast division of The Computer Museum. The focus is entirely on the collection, preservation, display, and study of the history of computers and computer technology. The project is spearheaded by Gwen and Gordon Bell, who founded the original museum in Boston, and Leonard Shustek, cofounder and fellow of Network General Corporation who serves as chair of the History Center, but it will depend on the continued enthusiastic support of both volunteers and donors to make it a success.

### THE AMAZING COLLECTION

At the core of the Center is the museum's existing 15-year-old collection, which is already the world's most comprehensive archive of computing artifacts and memorabilia. This historical collection, most of which has not been on display in Boston, contains 1500 artifacts, 2000 photos, videotapes and films, and an extensive technical documentation library. Artifact highlights include the 1951

Whirlwind, the 1952 UNIVAC 1, the PDP-1 with the original SpaceWar simulation, the famous ILLIAC IV multiprocessor, and most of Seymour Cray's computers from the 1957 NTDS-17 built for the Navy to the 1976 CRAY-1 to the ill-fated CRAY-3 "shoebox" processor.

Many of the items, like the Whirlwind and the ILLIAC IV, are one-of-a-kind historical pieces that could not possibly be duplicated in any other collection, but we also have an extensive collection of more recent mini-computers and personal computers such as PDP-10, Xerox Star, Altair 8800, and Apple I. We keep duplicates of important artifacts that are not unique so that we can mount multiple exhibits or loan pieces to other institutions. We are the only computer museum to have a joint collections agreement with the Smithsonian, and we have a good history of cooperation with the Babbage Institute.

Right now the collection is aggressively being expanded with both new and old computers. In the last hours of the last day of 1996—so that the owner could get a tax deduction that year—we accepted the donation of an IBM 7030 STRETCH supercomputer from 1959. It is serial number 5 of 7 ever built, weighs 10 tons, and was the machine that produced the technology for both the 7090 and 360 series. Except for one which might still be somewhere within IBM, we believe it to be the only complete STRETCH which still exists. Other recent additions include the hand-wired prototype of the Busicom calculator for which the first 4004 microprocessor was designed, the original MIPS microprocessor wafer manufactured at Stanford University, and the Robert Morris "Internet worm" software. We often collaborate with other institutions for preservation of valuable material; for example we are currently working with Stanford on joint accession of a large assortment of personal computer software and hardware.

The collection is the major asset of the history center and will be available for the production of exhibits and publications as well as for use by scholars, educators, researchers, engineers and journalists.

## Background: THE COMPUTER MUSEUM

The Computer Museum is a financially-stable 501(c)(3) non-profit institution dedicated solely to computers and people and their impact on each other. It owns a 53,000 square foot facility in Boston, including 7 exhibition galleries and an auditorium, which two million people have visited in the last decade. The annual budget is about $4M, of which about 45% is earned and the remainder is contributed.

The museum's international reputation has enabled it to assemble a distinguished board of 25 trustees and overseers, including 10 leaders from the computer industry on the west coast. The board has enthusiastically endorsed the creation of the new History Center. The east coast facility will continue its current mission of providing interactive hands-on educational exhibits directed primarily at children and the general public.

The west coast facility will be the center for the historical collection and display. The public exhibits here will be oriented towards adults, with "Scientific American"-level explanations. An important aspect of the Center will be programs for study and research into the history of computer technology, for which we have established a strong connection to existing projects at Stanford University, and we expect to continue as allies and collaborators.

## CURRENT ACTIVITIES

The Computer History Center has offices within the Ames Technology Commercialization Center in Santa Clara, which is a high-tech incubator partially funded by NASA. The warehouse for storing the collection is on Moffett Field in Mountain View. About half (120,000 pounds) of the collection that was not on display in Boston has already been transported to the warehouse. We have four full-time paid staff, including a collections manager who is also a Ph.D. candidate in the History of Technology at Stanford. We have a monthly "volunteer weekend work day" at the warehouse when, by prior arrangement, friends of the History Center can come to Moffett Field to help with sorting, cleaning, arranging, documenting and moving artifacts.

As part of initial activities we staged an exhibit using artifacts from the collection for a celebration of the 25th anniversary of the microprocessor at the recent MicroDesign Resources *Microprocessor Forum* in the San Jose Fairmont. Jointly with

MDR we created a poster detailing the history of the microprocessor, which was previewed at the Forum and will be printed in February. We similarly created, along with Intel and Softbank, a major 5000 square foot microprocessor and personal computer exhibit at the recent Comdex exposition in Las Vegas.

We are currently working on a diverse series of historical exhibits for installation in the new Gates Computer Science building on the Stanford University campus, and on a major display for the 50th anniversary of the ACM in San Jose in March 1997.

Until we have our own facility we will continue to create these kinds of exhibits for deployment at conferences, public spaces, and corporate lobbies. To further public involvement we are also scheduling a lecture series and film series on topics in the history of computation.

The budget for the center is currently about $250K/year, which is provided by a combination of funds from the existing museum budget, operating-fund donations of $10K to $25K/year each from supportive companies, and founding member subscriptions of $1K to $5K each.

### FUTURE PLANS

We are in the process of planning for a future permanent public facility in Silicon Valley to house the collections, exhibits, research and administrative offices. It could be an existing building or a new building.

The most likely scenario is a new 60,000 square foot facility built on land made available to us by a city government or a university, and we are having the appropriate discussions along those lines. For that purpose we have planned to begin a $30M capital fundraising campaign, half of which would pay for the building and half of which would provide an endowment necessary to make the Center self-perpetuating. We have already raised $2.2 million for the capital fund.

One third of the building (20,000 sq. ft) would be dedicated to permanent rotating exhibition space and would serve as a public attraction. Another third, probably below ground, would be for dense but accessible storage of the rest of the collection not on display, available to researchers, educators, writers, and lawyers. The final third would be used for research and administrative offices, an auditorium, a library, and exhibit preparation areas. The entire collection is available to support exhibits and publications as well as for use by scholars, educators, researchers, engineers and journalists.

The focus of operations will be the three missions of the History Center: (1) preservation of artifacts in all forms—hardware, software, documents, photos, and histories, (2) public exhibits, both at the Center and over the Internet, and (3) research and study projects. The latter will include a variety of academic and scholarly activities, including the development of a set of courses on the history of computers, and investigations into appropriate techniques for the preservation and historical analysis of software and the Internet.

The Computer History Center is in operation now, and is making good progress towards the ultimate vision. The achievable goal is to make this the premier international resource for the history of computers. Silicon Valley and the computer industry deserve no less. Join us!

*Len Shustek, co-founder of Network General Corporation, is chair of The Computer Museum History Center Project and a board member of The Computer Museum. He can be reached at shustek@ngc.com.*

## CORRECTION

On page 43 of ENGINE 3.3, in Roy Allan's article "What Was The First Personal Computer?" the Kenbak-1 is referred to as "microprocessor-based." Of course it wasn't, since it predated the Intel 4004, but the mistake arose from editing and is in no way the fault of Mr. Allan.

## PUBLICATIONS RECEIVED

Amateur Computer Group of New Jersey NEWS.

Volume 21, Number 2, March 1996. Buying a PC; Checking your drives; Happy Computing; Win95; Software reviews.

Volume 21, Number 3, April 1996. Missing from archive.

Volume 21, Number 4, May 1996. Missing from archive.

Volume 21, Number 5, June 1996. LaserJet 4 Problems; Reinstalling 95; UNIX FAQ; more.

Volume 21, Number 6, July/August 1996. Missing from archive.

Volume 21, Number 7, September 1996. Missing from archive.

Volume 21, Number 8, October 1996. $400 Computer; Convention Report; Software reviews.

Volume 21, Number 9, November 1996. In Defense of Older Computers; Realaudio; Investing SIG; Software reviews.

Volume 21, Number 10, December 1996. Pilgrim's Search; Horsing Around with Standards; The Useless Editor; Elections; Software reviews.

Volume 22, Number 1, January 1997. Learning Visual Basic; Safe Computing; Internet Corner; Software reviews.

Australian Computer Museum Society Newsletter

#12, August 1996. History of Microsoft.

#13, October 1996. History of Information Services, Commonwealth Bank of Australia.

#14, December 1996. History of Microsoft continued; Mastertouch Piano Roll Company.

Charles Babbage Institute NEWSLETTER.

Volume 18 Number 1, Fall 1995. 50th Anniversary of ERA; CBI Staff Changes; ENIAC plans.

Volume 18 Number 2, Winter 1996. Stephen Johnson New Associate Director; ENIAC Extravaganza; Integrating Archives with DEC. From Bruce Bruemmer.

The Computer Journal, Issue #79, Fall 1996. PC Serial Port in FORTH; AT Modem Commands; The P112 Z182 Board; C and Assembler; more. $24/6 issues, $44/12 issues. From Bill Kibler.

The Computer Museum NEWS. Winter 1996. Kids' Software Library; Spring 1996. Website; Update; Summer 1996. Kids' Software Library.

[The Hewlett-Packard Journal will be logged in a future issue.]

International Calculator Collector, journal of the International Association of Calculator Collectors.

Issue #12, Spring 1996. CURTA; Red Calculators; HP Handheld Conference; Book on Sinclair; classifieds, resources, more.

Issue #13, Summer 1996. Pulsar Calculator Watch; TI Calculators, 1972-1979; Boston Computer Museum; Price Guide Update; classifieds, resources, more.

Issue #14, Fall 1996. Commodore Calculators; World's First Electronic Calculator; Membership directory; Calculator Web site; photo gallery, classifieds, resources, more. US$16 per year with membership ($20 foreign). From Guy Ball.

Random Output, newsletter of East Bay FOG.

Volume 12 Number 2, February 1996. Good Tech Support; What Use is the Internet?

Volume 12 Number 3, March 1996. What Use is the Internet?; HP LaserJet.

Volume 12 Number 4, April 1996. Missing from archive.

Volume 12 Number 5, May 1996. E-mail; Windows Problems.

Volume 12 Number 6, June 1996. Missing from archive.

Volume 12 Number 7, July 1996. Missing from archive.

Volume 12 Number 8, August 1996. Free E-Mail; Back to Basics.

Volume 12 Number 9, September 1996. Back to Basics; Gale Rhoades.

Volume 12 Number 10, October 1996. Boston Computer Society Closes Doors.

Volume 12 Number 11, November 1996. Smaller Bytes; AOL Gets Smart.

Volume 12 Number 12, December 1996. Annual Party; Virus Directory.

Volume 13 Number 1, January 1996. Intuit's QuickBooks. 4 pp. From Pete Masterson.

The Z-Letter, newsletter of the CP/M and Z-System community.

Number 40, November/December 1995. Community news; Lambda Access; MYZ80 part 2; S-100 treasure trove; resources, publications, letters and classified. 20 pp.

Number 41, January/April 1996. CIS COBOL; GSX-80 1.1; Kasparov beats Deep Blue; Konrad Zuse; 50th anniversary of ENIAC. 20 pp. Last issue.

## ADDRESSES OF CORRESPONDING ORGANIZATIONS

Amateur Computer Group of New Jersey (ACGNJ), P. O. Box 135, Scotch Plains NJ 07076. Joe Kennedy, president.

Australian Computer Museum Society, PO Box 103, KILLARA 2071, NSW, Australia. Michael Chevallier, secretary.

Charles Babbage Institute, 103 Walter Library, 117 Pleasant Street SE, Minneapolis MN 55455. Bruce Bruemmer, archivist.

Commercial Computing Museum, 220 Samuel Street, Kitchener ON N2H 1R6, Canada. Kevin Stumpf, president.

Computer Conservation Society, 15 Northampton Road, Bromham, Beds. MK43 8QB, UK. Tony Sale, secretary.

The Computer Museum History Center, Box 3038, Stanford CA 94309-3038. Dag Spicer, collections manager. Note change of address and contact.

The Computer Journal, P. O. Box 3900, Citrus Heights CA 95611. Dave Baldwin, editor.

Computer Preservation Society (Inc.), Ferrymead Historic Park, 369 Bridle Path Road, Christchurch, New Zealand. Abraham Orchard, secretary.

East Bay FOG, 5497 Taft Avenue, Oakland CA 94618. Tom Lewis, president.

Hewlett-Packard Journal, Hewlett-Packard Company, Box 51827, Palo Alto CA 94303-0724. Richard P. Dolan, editor.

Historical Computer Society, 2962 Park Street, #1, Jacksonville FL 32205. David Greelish, president.

International Association of Calculator Collectors, Box 345, Tustin CA 92781-0345. Guy Ball, Bruce L. Flamm, directors.

IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos CA 92640. Bob Carlson, director.

Lambda Software Publishing, 149 West Hilliard Lane, Eugene OR 97404. David A. J. McGlone, editor and publisher.

Lexikon Services, Box 1328, Elverta CA 95843. lexikon2@aol.com. Mark Greenia, director.

Perham Foundation, 101 First Street #394, Los Altos CA 94022. Don Koijane, president.

San Francisco Computer Museum, Box 420914, San Francisco 94142-0914. Erich W. Schienke, manager.

Santa Clara Valley Historical Association, 525 Alma Street, Palo Alto CA 94301. John McLaughlin, director.

## NINES-CARD....

will be back next issue. And we don't want anybody telling us that the computer industry has run out of funny stories!


## The ANALYTICAL ENGINE

**Volume 3, Number 4, Fall 1996**
**ISSN 1071-6351**

journal of the Computer History Association of California, is published four times a year at Palo Alto, California. Basic, domestic subscriptions are $35 paper, with $25 deductible as a charitable donation. Inquire about institutional, international, and low-income subscriptions at:

**4159-C El Camino Way**
**Palo Alto, CA 94306-4010 USA**
**Internet: engine@chac.org**
**WWW: http://www.chac.org/**

# CONTENTS


Editorial: NOT EASY, BUT FUN .... 1
YES, ADS .... 2
TZL EOF (sigh) .... 2
SOURCE TO THE PEOPLE! .... 3
VIRTUAL VARIETY.... .... 3
CALCULATOR CATALOG! .... 4
THE IMPORTANCE OF SOFTWARE HISTORY by Luanne Johnson .... 4
THANKS TO .... 5
NEXT ISSUE / COVER ART .... 5
MICROPROCESSOR AND MICROCOMPUTER: Invention and Evolution, by Stanley Mazor .... 6
UP ON THE RIDGE: The Story of Ridge Computers as told by John Sell .... 13
IN MEMORIAM: CUTHBERT C. HURD .... 21
IN MEMORIAM: SEYMOUR CRAY .... 24
DID YOU HEAR ANYONE SAY GOODBYE? The End of Atari by Don Thomas .... 26
BAGGING A JAGUAR .... 29
Book Review: WHERE WIZARDS STAY UP LATE by Katie Hafner and Matthew Lyon, reviewed by Erich Schienke .... 29
SPOTTER FLASH .... 31
THE COMPUTER MUSEUM HISTORY CENTER by Len Shustek .... 31
CORRECTION .... 33
PUBLICATIONS RECEIVED .... 33
ADDRESSES OF CORRESPONDING ORGANIZATIONS .... 35
NINES-CARD.... .... 35

US$5.00 UK£3.50 10DM ¥700

CHAC, 4159-C El Camino Way, Palo Alto CA 94306 USA FIRST CLASS MAIL

# HELP!

The San Francisco Computer Museum (SFCM) and the Computer History Association of California have been offered a substantial, historically significant accumulation of (mostly micro) computer artifacts, software and documentation located approximately 80 miles from San Francisco, in the heart of tomato country.

This collection is remarkably diverse (read: treasure, junk, and everything in between,) barely sorted, and thoroughly dusty. It is in **paid** storage, so the sooner we remove it, the better off we are. For this rescue we need, **as soon as possible:**

- **A truck.** The bigger the truck, the fewer round trips will be necessary.
- **Volunteers** who can handle artifacts — mostly boxed, some heavy — and have experience with sorting computer equipment. Expert supervision will be provided.
- As always, **money!!** Rescuing this material and transferring it to inexpensive storage is a much better deal than continuing to pay for the current site; but accelerating the project means that we need cash fast.

**Join this rescue and see micro stuff you've scarcely even dreamed of.** This is a once-in-a-lifetime experience. *You will need good tolerance for heat and dust.* Intrepid rescuers please e-mail engine@chac.org or erich@fog.com immediately, or call 415-703-8362. **Thanks!**

## San Francisco Computer Museum / Computer History Association of California