

The *Analytical Engine*

JOURNAL OF THE COMPUTER HISTORY ASSOCIATION OF CALIFORNIA



The Analytical Engine

JOURNAL OF THE COMPUTER HISTORY ASSOCIATION OF CALIFORNIA

Editorial: THREE YEARS

On April nineteenth, with only the most minor ceremony, the Computer History Association of California observed its third anniversary. We look back on three years — and especially on the most recent year — of substantial accomplishment and profound satisfaction. This is a personal, and necessarily abbreviated, report on the current state of the CHAC's projects and prospects.

Publication: You're reading the tenth issue of the ANALYTICAL ENGINE, and we look forward — with you, we hope — to many more. Its present situation has many positive aspects: Good material is coming in unsolicited. We have several prospective subjects for interviews, with at least the next four solidly planned; preservation of oral history is and will continue to be the ENGINE's single most important mission. Volunteers have begun to share editorial responsibility. The ENGINE is poised to become the first general-circulation trade magazine of computer history in the United States.

On the other hand: This is ENGINE 3.3, which says May 1996 on it and should have appeared then. Since it's September as I write this, you might reasonably expect to receive issue 4.1; so the magazine has slipped by two full issues. Most of the discrepancy has opened up in the last few months. We, and you, want to get the ENGINE back on track, but it's proving difficult.

Why? Because, in the three years since our founding, computer history has attracted much more general interest — so that the CHAC staff finds itself handling hundreds of pieces of e-mail, firing off faxes to far corners, giving lectures and videotaping them, and *trying* to stay in control of a bulging inventory of hardware, software and docs. Just answering a single day's e-mail for the CHAC can take two to three hours of one person's time.

We're delighted to be so popular. But something — maybe it's living in Silicon Valley — makes us *acutely* aware that the CHAC may become a casualty of its own success. This organization has not grown in step with the interest in what it does. Later on we'll tell you about some new strategies to keep the CHAC on top of its own situation.

What else *haven't* we done? We haven't managed to publish much history from Southern California, in spite of making inquiries for about a year. As we inventory, we learn more about our SDS 930, which was built in Santa Monica; but we've published very little about the ARPAnet, not much about the fascinating computer culture that centered on UCLA, and nothing at all about computer use in the aircraft factories, or the development of the many experimental "one-off" mainframes. Attracting this material will continue to be a high priority.

We've published nothing so far about military computing, certainly a rich topic in California above all. One or two articles now in preparation may remedy this before the end of the year.

The sparse selection of powerful desktop publishing software for the MS-Windows platform is irritating. The ENGINE, now produced in Microsoft Word, is about at the limit of what that very capable word processor will accomplish easily. Our plans for ad layouts, more and better illustrations, and color will soon make more advanced page composition imperative.

The electronic edition of the ENGINE has slipped badly, for reasons too gory to relate; profuse apologies to those subscribers who are waiting for it! 3.1, 3.2 and succeeding issues will appear in electronic format as soon as we can produce Acrobat .PDF files. Andrew Eisner's generous gift of a Macintosh SE/30 may help bridge this gap; for the latest developments, pay an occasional visit to the ENGINE's Web page, <http://www.chac.org/chengin.html>.

Collection: Thanks to our newest collaborators and benefactors, the Perham Foundation of Los Altos CA, the CHAC now has a modest amount of storage space that will serve us well until the Computer Institute/SFCM is established in San Francisco — see p. 47. Accordingly, we are once again acquiring software, docs, and hardware. Read this issue's Acquisitions column for the juicy details on, for example, our new HP 2114A.

Recently we have committed significant time, energy and money to improving utilization of storage. Edwin V. El-Kareh, the CHAC's new Tactical Director, is working energetically to replace all cardboard cartons with lidded plastic crates. These not only provide much better protection for artifacts, but they are stackable, meaning that our storage areas can now be used literally to the roof. Over the summer we'll also be scrounging steel shelving. Now that the founding of a computer museum in the Bay Area is a likely prospect, your Association's role as a conservator has been revitalized.

MUSEUM ACTIVITY

Closest to our hearts, of course, are the two computer museum projects now taking shape in the San Francisco Bay Area. The Computer Institute in Berkeley, headed by Fred Davis, Sylvia Paull, and Andrew Eisner, plans a museum as one part of a "global center of computer technology" in downtown San Francisco; see Fred's article on p. 47 for the ambitious details. A second effort, the Computer Museum History Center, is being planned by Gwen and Gordon Bell of the Computer Museum in Boston MA; we are promised an update on this project for ENGINE 3.4.

Personally, we hope that these two projects find a way to converge if not actually combine. As Erwin Tomash has wisely suggested, "Of all ways to teach history, a museum is the most expensive;" and philanthropy in California, as in most places, has long passed the lighthearted grip-and-grin stage. The Bay Area's computer museum, to justify its substantial cost, *must* offer commensurate value — both perceived and real — in education and in entertainment. Building such a museum, and even making the best case for it, will require every scrap of energy that all the interested parties can summon up together.

Heinz Nixdorf Museumsforum

Energy, meanwhile, seems abundant at the new Heinz Nixdorf Museumsforum in Paderborn, Germany. Curators Ulf Hashagen and Dr. Karlheinz Wiegmann are building on enviable foundations — a realistic budget, a deep fund of knowledge, and *serious* floor space in an existing multistory building. These assets, together with their obvious dedication and the luck they richly deserve, will soon produce one of the finest computer museums in Europe. See page 46.

Australian Computer Museum Society

The Australian Computer Museum Society has circulated a Site Planning report and proposal for its long-contemplated facility. The authors come to many of the same conclusions as has the CHAC:

A considerable amount of capital and resources, including volunteer labour, would be necessary to establish and maintain a Computer Museum to attract a significant public interest.... Because of the planning time for major developments....[t]hree to five years would be good going, but the 2000 Olympic year is a good target. If established as part of a major development this strategy will secure us a good site in a prime location.... We need, in the immediate future, a site that we can use as a workshop to actively engage in restoration and development work. We also need storage space, preferably integral with the workshop.

[ACMS Newsletter #10, April 1996.]

We believe that, like the CHAC, the ACMS is destined to "ride the wave" of public interest to a resounding success. The recent, well-publicized purchase of a module of Babbage's Difference Engine by the Power House Museum in Sydney is a clear indication that computer history has become a newsworthy topic in Australia. In the last year the ACMS has raised \$Aus10,000 through a direct-mail campaign and begun site negotiations with the Australian Technology Park in the Eveleigh workshops at Redfern, Sydney.

On the other hand, ACMS President Graeme Philipson laments "two major problems....the continued inability of the Society to find a suitable storage area....[and] lack of time." Ain't it the same the whole world over!

IN MEMORIAM: DAVID PACKARD

David Packard, who co-founded one of the world's most highly regarded businesses and personally guided it to undisputed leadership in its field, died in Palo Alto, CA, USA on March 26, 1996. He was eighty-three.

Packard managed the Hewlett-Packard Corporation in a deeply personal and proactive way from its founding in 1939 to his retirement in 1993. In the process he set an example for technically adept, skill-conscious companies throughout the world; he also helped build a company which, in 1995, had over 100,000 employees and annual revenue of \$US31.5 billion. HP is routinely cited for its enlightened treatment of personnel, its innovative management, and its unrelenting pursuit of quality, all principles which Packard believed fundamental to corporate survival and growth.

Packard was born in Pueblo, CO, USA, and as a boy was interested in electricity, general science, and particularly radio engineering, building a "fairly sophisticated vacuum-tube receiver"¹ when he was twelve and becoming the secretary of his local radio club in high school. At sixteen he operated his own ham radio station, 9DRV. His other favorite pursuit was riding, a love and skill that he kept up almost to the end of his life.

In the fall of 1930 Packard enrolled at Stanford University, and met his future business partner and lifelong friend, Bill Hewlett. Packard, Hewlett, Ed Porter, and Barney Oliver were all encouraged — primarily by the legendary EE professor Fred Terman — to pursue careers in electronic engineering, a field then so new that it was assumed to be an ill-defined subset of "radio." Packard also distinguished himself as an athlete, setting records in track and earning varsity letters in football and basketball.

Terman's four star pupils, in the spring of 1934, determined to start "something on their own" using their cutting-edge technical training, which Terman had carefully reinforced with hands-on experience. But the founding of the new enterprise

had to be deferred when Packard received a job offer from General Electric — a prized commodity in the depths of the Depression.

General Electric, however, was not poised to take best advantage of Packard's talent. The company had no interest in electronics, and assigned him to "uninteresting" work in refrigeration. He quickly wangled his way into other positions, and found rewarding work troubleshooting GE's production of vacuum tubes. The yield of one particular mercury-vapor rectifier, a large and expensive component, was disastrously low. Packard responded by spending "most of [his] time on the factory floor," working closely with line employees to pinpoint and remedy causes of tube failure. The effort was a resounding success and Packard's first application of "management by walking around," which he soon recognized as key to any kind of technical manufacturing.

In 1937 Packard briefly visited Palo Alto and held the first recorded business meeting of his "proposed business venture" with Bill Hewlett. But his employment at GE was so important, at a time when jobs were still brutally scarce, that he returned to upstate New York immediately; and his bride-to-be, Lucile Salter, traveled there by train (a four-day journey) so that the two could marry and have their honeymoon in barely more than a weekend! Packard would not formally resign from GE for years to come.

In the meantime, the Packards were at least able to settle in Palo Alto, thanks to a Stanford fellowship and stipend arranged by Terman. Packard worked with other electronic innovators, including the Varian brothers and Charlie Litton, on advanced development including vacuum-tube engineering, a field still of considerable interest to GE. But in 1939 he and Hewlett were at last able to start their "venture" in the famous garage on Addison Avenue, which would be declared a California Historic Landmark fifty years later.

Some of Hewlett and Packard's earliest projects, in the search for income, ranged far afield even for H and P — a harmonica tuner, a telescope tracking controller, even a foul-line spotter for a bowling alley. But the first "Hewlett-Packard product" was an audio oscillator, well-designed and priced far below its competition....actually, and accidentally, priced below cost. Packard christened it the Model 200A to give the fledgling company an air of expe-

¹ *The HP Way*, David Packard, HarperCollins, 1995, p. 7.

rience. Within months the oscillator had been improved, and the resulting Model 200B sold well at the (corrected and profitable) price of \$71.50 each! Eight 200B's were sold to the Walt Disney Company for use on the movie *Fantasia*. By the end of 1939, HP's first full year in business, the company recorded a respectable profit and moved to larger quarters on Page Mill Road.

World War II, justly called the "electronic war," saw HP expand its facilities again and put line workers on double shift. A perpetually over-extended Government was grateful for product quality combined with timely delivery; HP became one of only three California companies to earn the coveted Army-Navy "E" award. Although wages were frozen in wartime, HP's pre-existing bonus plan — coupled with remarkable gains in productivity — made it possible to pay bonuses of as much as 85 per cent of base wage. Tangible recognition for improved productivity became another cardinal principle of HP management, one that accorded with Packard's deeply philanthropic nature.

When peace came and controls on employment were loosened, Packard gave his corporate strategy unfettered expression. Any company's greatest asset, he insisted, was a highly skilled, highly motivated, and meticulously trained employee. The company, therefore, had three primary obligations; the first was to hire top talent, the second was to assure congenial and secure conditions of employment, and the third was to maintain channels through which any employee could easily contribute to the company's success.

These strategic goals were put in place through tactics that were innovative and even startling for their time. Elaborate provision for the welfare of a company's workforce *per se* was nothing new; IBM, to take a well-known example, was renowned for the scope of support it offered to its employees. But whereas IBM intended that the individual worker should draw strength *from* a highly prescribed and strongly hierarchical corporate structure, Packard took the opposite tack, trying to restrict the "company" to no more than would serve to bind its employees into a productive relationship. Designed for agility and constantly pruning anything superfluous, Hewlett and Packard's HP was a company created by engineers for engineers

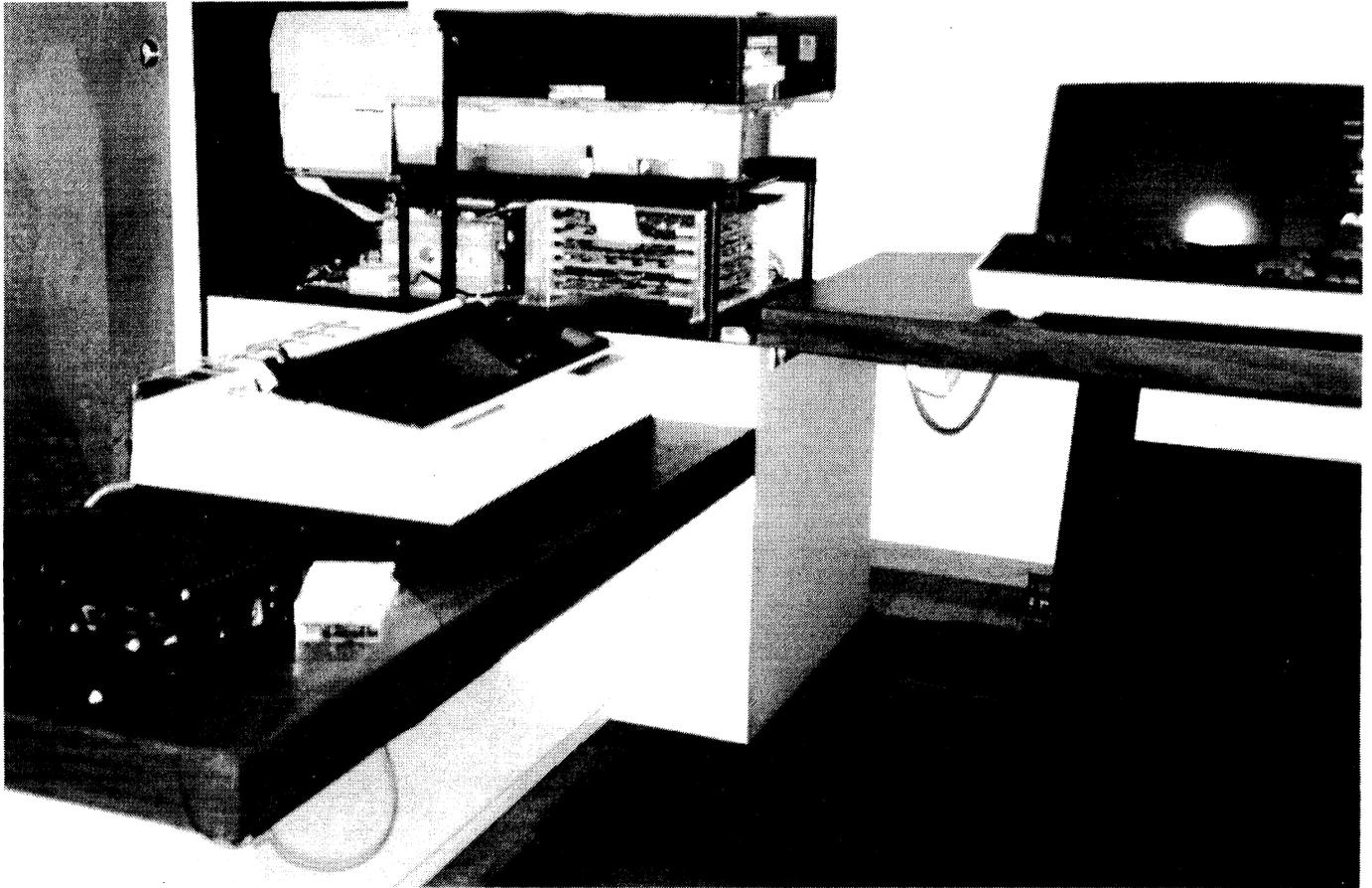
— and it worked; the strength of the company was the strength of its people.

Every effort was made to keep the company's structure "flat" and lines of communication short, so that executives with decision-making power would have ready access to the company's fund of ideas. Coffee talks, company picnics and barbecues, and open offices all contributed to the exchange of information; at the same time, a support structure that included flexible scheduling, tuition sharing, and catastrophic medical coverage made it clear that HP employees were a valued resource.

The HP Way, as a corporate culture, just about wrote the prototype specification for Silicon Valley's high-tech business. Perhaps this model now goes too far at times, creating personalities who are better-known than the companies for which they work. But when we look at HP itself, it's clear that the model worked to near-perfection at least once. *Si monumentum requiris...* look around and you'll probably see the HP logo!

David Packard would have been famous enough, certainly, if we had only HP itself to remember him by. He himself preferred to be remembered for his great philanthropic endeavors — the David and Lucile Packard Foundation, now the largest charitable private foundation in the United States; the Monterey Bay Aquarium, which has attracted millions of visitors to the Central California coast; Lucile Salter Packard Children's Hospital, one of the world's most highly regarded pediatric facilities; and visionary projects with a worldwide reach, such as the Center for the Future of Children and the Packard Humanities Institute. He was more than merely a generous man, more than a captain of industry, and more than a statesman. He was, at last, someone who understood the supreme good fortune of his lot in life, and worked hard to give as he had received.

The Computer History Association of California extends condolence to Mr. Packard's children, David W. Packard, Nancy Ann Packard Burnett, Susan Packard Orr, and Julie E. Packard, and to his many colleagues and friends throughout the world.



The LOMAC ADAM (1977) with 50Mbit Control Data hard disk raised for service, terminal, and Centronics dot-matrix printer.

THE COMPUTER WILL DO ANYTHING YOU TELL IT TO DO: An Interview with Maiga Wolf

HLC: See, in accounting, the "assets equals liabilities plus equity" -- that was from about 1300. So, from 1300 to 1967, huge companies -- international companies -- functioned without computers in their accounting departments. Then suddenly, from 1967 to now, nobody can even write a check without a computer. What I'm saying here is that the computer has made a tremendous difference in everyone's life -- but it seems to me that you techies think you're the only ones who really appreciate what an incredible breakthrough computers are. I want you to understand how significant it is for the rest of us. I think you should explore how computing changed the life of an ordinary bean-counter.

KC: On January 19th we got a call from Maiga Wolf. She had an ADAM minicomputer that had

been built by Logical Machines Company in Sunnyvale, and she was retiring from twenty-plus years of running her own bookkeeping business; would the CHAC be interested? The software, oh, easy! She wrote it herself.

So Hilary Crosby, a certified public accountant; Edwin El-Kareh, an engineer and CHAC volunteer; George Durfey, an engineer and photographer, and I rose to this challenge.

SYSTEMS MATTER

EVEK: You worked in a large company in the accounts payable department?

MW: I was brought in to revise their accounts payable. They had a big problem. They couldn't pay their bills in time, they couldn't get discounts in time, and it was one big mess. This was in 1963.

HLC: And they had no automation in the accounts payable?

MW: No, everything was manual. I had 29 people in accounts payable, processing bills for Ampex. I went in and revised the procedure, how they were handling [paperflow.] In the first year I managed to recapture \$275,000 of discounts. In those days that was a lot of money. [To begin with] we didn't get all the discounts. It took me a while to get there, but by the end of the first year we had saved that much.

In Canada originally I was working for a company, but on the side I was trouble-shooting. The other chartered accountants would come to me and say, "Oh, one of my clients has so many problems, can you come over and straighten them out? We don't have time." So I'd work at that site on Saturday or Sunday. Finally I had so many waiting for me that I realized I was earning more doing consulting than from my regular job. So I quit, and began arranging systems to process paperwork, full-time.

HLC: Were you using bookkeeping machines at that point?

MW: Not bookkeeping machines. There were Burroughs posting machines and calculators.

HLC: Did you have your accounts receivable integrated with your sales and the manual system?

MW: No, no. I had separate girls doing separate jobs. One girl would do the invoicing, and the other girl would do the recording in the sales journal, and another girl would take the invoices and post to the receivables.

HLC: So each invoice got posted twice — once in the sales journal, and once as accounts receivable to the specific customer?

MW: Right, manually. Then in 1967 or '68, I don't recall the exact date, when we went on the computer at Precision Instruments, it went from the sales journal right into the accounts receivable. They had the payroll in a bank, and that was also computerized, but the bank had lots of problems there too.

HLC: So, by the late sixties, all the tax withheld and the deposits and the quarterly reporting was handled by the bank on a computer, and people just got their paycheck, and at the end of the year their W-2?



Maiga in 1947.

MW: Well, I'm not a certified public accountant in California. I studied in Canada to be a chartered accountant, and I went into accounting for one reason. I was an immigrant after the Second World War and I could see that I wasn't going to do anything with the positions that were available for young women in '48, '49 -- I didn't want to be a nurse, I didn't want to be an office clerk or a teacher -- that was about my choice, and it just didn't appeal to me. So I decided there was a little bit more leeway if I went into accounting, and so I studied in London, Ontario.

First you got your four years at the university. Then you had to be an apprentice with a chartered accountants' firm, the equivalent of a CPA firm, for five years, and then you wrote your exam and got your papers.

When I moved to California, I wanted a job to start with, and Ampex hired me as a consultant in their systems and procedures -- they had problems, especially in accounts payable. I went in there and started to work, and things just rolled along, and I never got around to doing my CPA because nobody asked for it. [Laughs.]

HLC: What about tax returns?

MW: I never did tax returns. After I quit Ampex I wanted to see what the tax laws were in California, so I hired myself in with a small CPA firm. And I just happened to be there between January and April, with the tax returns and stuff, and I said, "That's a rat race, I don't want that." And that's why I stuck to this [bookkeeping] so I never had to bother with taxes. Most of the small businesses also have personal stuff involved, and I don't want to know anything about the personal.

MW: I didn't have to worry about those. That belonged actually in personnel, so I didn't have much to do with the payroll at all, except to take the feeders [reports] and put all that in the general ledger.

HLC: When was the first time that you encountered the computer?

MW: When I was a controller at PI [Precision Instruments] in Palo Alto. They decided to install a computer, one of the great big ones with punch cards, and my accounting department posted in through the computer.

KC: Was this an early minicomputer or a mainframe proper?

MW: It was a mainframe, I think a big Burroughs. I wasn't familiar with computers at all, so whatever the company brought in, that was it — I had no say as to what type. All the departments were throwing stuff at the programmer, and expected him to have answers for whatever came into their heads.

KC: Meaning that the department heads expected the programmer to know their business and requirements.

MW: Right, and it didn't work, of course.

KC: Never does.

MW: And so I was looking at what they were doing, and I realized that nobody told the programmer what they actually needed [in output] from that machine. The engineering, and marketing, and other departments threw everything at the poor programmer and said "Do it." And it would come out wrong, or not work at all, and they had lots of problems. So I decided, that's foolish, this poor fellow is really not an engineer, or marketing manager, or whatever else they expect him to be. When my turn came to put the accounting in, I broke it into separate phases. I put in — let's say — sales first, and got the sales journal, then I put in cash receipts, and so on. By working with one phase after the other, we brought it off with only slight adjustments. Everybody [else] was hollering "How come you don't have anything wrong when we all have problems?" and they were cussing the programmer! And I said, well, you want him to bite it off and chew it up for you, and he doesn't know what you need. You have to tell him what you need.

I worked for Precision Instruments something like 6 or 7 years. That was my start with computers, and the only experience I had with computers, until I bought my own.

ALONE, AT LAST

KC: And you went out to do more systems installation?

MW: No, I began doing bookkeeping for companies that were too small to keep an in-house accountant or a full-fledged bookkeeper. I got into that in a strange way — maybe not so strange, considering what had happened in Canada. Someone I had known at PI, not from my own department, called me up and said "I have two good friends who want to buy into a business — " which was a bar and a restaurant in Palo Alto — "but the fellow hasn't got any books. Is there any way you can help?" I'd just quit PI and was sitting at home, thinking about what to do next; so I said that I'd take a look.

I had no equipment at home. My neighbor, who had a typewriter [that was] from her club, said "I'll let you use my typewriter." My insurance guy had just bought a new calculator, so he gave me his old calculator. At first I wasn't going to buy anything, because I didn't think I would continue that way. I set up all the books for the restaurant. It was early in the year, something like March or May, so it wasn't too hard to catch up. I took the 1120 [corporate tax return] balances forward and set up the new year. The restaurant looked like it was making money, and I told them, "That's what I can get out of the paperwork." Then, of course, this fellow who had the bar had a friend who was in trouble, and.... [Laughter.] It just mushroomed, so I got an adding machine and a typewriter of my own, and off I went.

HLC: What made you decide that in order to have the volume, you needed a computer — what put that together for you? How did you know that to service the number of bookkeeping clients that could give you an income, you needed a computer?

MW: I was on my own and I needed more income. Right away, I could see that I wasn't getting very far manually. I was spending hours and hours adding back and forth across all those pages in the ledgers and journals, and it was very time-consuming.

HLC: There weren't too many women in accounting -- or in computers -- at that time.

MW: Well there weren't too many [women] controllers at that time either. I was a controller for Precision Instrument.

HLC: Did you think people there resented you, a woman, having that job?

MW: Oh yeah, very much. The vice presidents from different sections would come — from marketing especially — and say "That girl in the office there, she wouldn't sign that...." Conrad Schoebel backed me up very well. He let them know, if Maiga said no, it meant no.

HLC: When you went to that convention in Pasadena, where you were looking at computers, how many other women were there as customers? Half of the people?

MW: No, very few, very few.

HLC: Did it feel strange, walking around there as a customer and a woman?

MW: No, because I've always been working in a man's world. In 1953, accountants were not women, and so I've been looked at — you know — as an oddball.

HLC: The guy from LOMAC that came to help you, the guy —

MW: Gary Kench, was very helpful [and] very nice.

HLC: Did he ever say anything like "You're good at this for a woman...."

MW: Usually he said something like "How's it going," and I would say "Well, it seems to be working," and he might say "I knew you would do it." That was about it.

KC: He was the sales guy for the company, right?

MW: Well, he was the salesman for this computer. What else he did for the company I have no idea.



Looking at the books.

Just at that time I read an announcement in the San Jose paper of the first Business Computer Conference, in Pasadena, with displays and the whole business. I got on the plane and went there, spent five days walking from place to place, from one computer to the next, and attending any lecture that was given. I came to the conclusion that *there wasn't one program* that would suit me! And I couldn't afford an in-house programmer, who at that time would have charged around \$40 an hour — not to mention that I'd probably have to wait two weeks before a freelancer would come, and I couldn't spare that kind of time either. Then I discovered this ADAM, and they assured me that I could program it myself, so I bought it.

KC: Those assurances have been given since there have been computers, and they haven't always been true, but in your case it was true.

MW: I had a good start. Gary Kench, the salesman, said he would come and get me started. In the beginning, he spent a couple of hours with me every day. Meanwhile I was studying the manual and trying to do the homework he gave me. Then the next time he came, if my work was okay, we just progressed from there. After about three or four weeks he said "You're on your own," and I took it from there.

KC: It's important to point out that this machine is not a micro in any sense. This is a minicomputer with a fourteen-inch disk and a terminal, and it's built into a desk along with a [dot-matrix] printer. If you don't mind my asking, how much did the whole shebang cost when you brought it home from LOMAC?

MW: Thirty-two thousand dollars.

KC: In 1977.

MW: Yes.

HLC: You had to keep a lot of books to pay off \$32,000 in 1977!

MW: It paid off very easily. As far as money went, it was no problem. But it was a big risk for me in the sense that I had no idea if I could handle the programming or not. There was no other way, so I took a chance and said that if it didn't work, it didn't work.

KC: And you had to buy one disk pack per person.

MW: Each customer had two [disk packs], one for backup.

HLC: So you kept a back-up in the same box?

MW: Yeah, they were labeled, number 1, number 2. I would take a back-up after every two hours' work just in case.

HLC: And how long did it take to do a back-up?

MW: Just a few minutes, I just wanted to copy the information from the built in disk.

HLC: How much were you paying for those disk packs?

MW: I have to think, now. At first I had only the two disks that came with the computer. Then as I started to work with separate accounts I needed more disks, and Gary offered me 20 disks at something like \$175.00 each. In my very first year I bought all those disks, because I knew I was going to enlarge.

KC: Did that also limit the number of clients you could take at one time?

MW: No, I could get more disks after that.

HLC: Are they still available?

MW: Yes, come to think, one of the computer supply catalogues had these large disks, still. Maybe not the same make — these are Control Data — but they had some hard disk [packs] still for sale.

KC: Just to recap: You had to computerize in order to get enough volume. You didn't find any ready-made software that would meet your needs, and you knew that you couldn't afford to invest time and money in custom programming from someone else; so you decided to do your own programming. Now, you were programming in BASIC?

MW: Well, in what LOMAC called English. I'm no expert on computer languages — I don't know anything about any other computer except this one, so I have no idea. They told me it could be programmed in English, and that's what I was doing.

HLC: So is your program actually a database where information is saved centrally, to be accessed from different perspectives and combined into reports?

0

GENERAL LEDGER

PERIOD FROM JULY 1, 1995 TO JULY 31, 1995

ACCOUNT NO.	PARTICULARS	PRIOR BALANCE	CURRENT MONTH	Y. T. D. BALANCE
110	CASH IN BANK			
202	111 BAY AREA BANK #1637701 2CR DEP. # 995 7.03.95 DEPOSIT		3 \$9,224.00	

GENERAL LEDGER

PERIOD FROM JULY 1, 1995 TO JULY 31, 1995

ACCOUNT NO.	PARTICULARS	PRIOR BALANCE	CURRENT MONTH	Y. T. D. BALANCE
526	INSURANCE-HEALTH			
440	3CK CK. # 663 7.12.95	CGLIC	2 \$218.90	
461	3CK CK. # 673 7.12.95	KAISER FOUNDATION	\$354.91	
628	3CK CK. # 15184 7.12.95	PACIFICARE	\$1,381.90	
630	3CK CK. # 15185 7.12.95	EMPLOYERS CHOICE	\$5,469.63	
669	3CK CK. # 15201 7.24.95	SAFEGUARD HEALTH PLANS	\$123.22	
893	5GJ DEP. # 995 7.03.95	THE EQUITABLE	-\$424.00	
ACCOUNT TOTALS		\$0.00	\$7,124.56	\$7,124.56

RECAP
 GL GROUP TOTAL PR is a verb,
 1 Does MOVE GROUP NO STANDBY
 2 and MOVE HEAD GROUP NO
 3 and IF GROUP COUNT < 2
 do GO TO NO TOTALS PRINT
 4 and PLF SLC (NPG)
 5 and PRINT "PR"
 6 and PSP 15
 7 and PRINT "-----"
 8 and PSP 15
 9 and PRINT "-----"
 10 and PSP 15
 11 and PRINT "-----"
 12 and PLF SLC (NPG)
 13 and PSP 26
 14 and JOIN STANDBY 0
 15 and PRINT 3 HEAD
 16 and PRINT "TOTALS"
 17 and PSP 16
 18 and PR\$ 15.2 PM TOTAL
 19 and PSP 12
 20 and PR\$ 15.2 MTD TOTAL
 21 and PSP 12
 22 and ADD PM TOTAL MTD TOTAL
 23 and PR\$ 15.2 SUM
 24 and LABEL
 NO TOTALS PRINT
 25 and PLF SLC (NPG)
 26 and PLF SLC (NPG)
 27 and MOVE 0 PM TOTAL
 28 and MOVE 0 MTD TOTAL
 29 and MOVE 0 GROUP COUNT

RECAP
 GL HDG is a verb,
 1 Does MOVE HEAD SAVE HEAD
 2 and TITLE PR
 3 and PSP 26
 4 and PXP
 5 and PRINT "GENERAL LEDGER" 0
 6 and PLF MLC 2
 7 and PSP 37
 8 and MOVE FISCAL YEAR BEGIN HEAD
 9 and PAD 6
 10 and CUT HEAD 4
 11 and JOIN TAIL HEAD
 12 and MOVE HEAD YEAR BEGIN
 13 and PRINT "PR"
 14 and PRINT "PERIOD FROM "
 15 and DATE PR YEAR BEGIN
 16 and PSP 4
 17 and PRINT "TO "
 18 and DATE PR END OF MONTH
 19 and PLF MLC 3 PSP 6 1
 20 and PRINT "ACCOUNT NO." 1
 21 and PSP 6
 22 and PRINT "PARTICULARS"
 23 and PSP 10
 24 and PRINT "PRIOR BALANCE"
 25 and PSP 14
 26 and PRINT "CURRENT MONTH"
 27 and PSP 14
 28 and PRINT "Y. T. D. BALANCE" 7
 29 and PLF MLC 2
 30 and MOVE SAVE HEAD HEAD

4 CASH RECEIPTS BATCH

JULY 1995

TRN	JNL	ACCT#	ACCOUNT	REFERENCE	DATE	PARTICULARS	AMOUNT
201	ZCR	121	ACCOUNTS RECEIV. - TRADE	DEP. # 995	7.03.95	MANAGEMENT, INC.	-\$6,300.00
202	ZCR	111	BAY AREA BANK #1637701	DEP. # 995	7.03.95	DEPOSIT	\$9,224.00

CHECK REGISTER BATCH

JULY 1995

TRN	JNL	ACCT#	ACCOUNT	REFERENCE	DATE	PARTICULARS	AMOUNT
460	JCK	111	BAY AREA BANK #1637701	CK. # 673	7.12.95	KAISER FOUNDATION	-\$354.91
461	JCK	526	INSURANCE-HEALTH	CK. # 673	7.12.95	KAISER FOUNDATION	\$354.91

```

RECAP
BATCH ENTRY is a verb,
1 Does IF BATCH COMPLETED = "NO"
  do GO TO INCOMPLETE
2 and JNL SELECTION-BATCH
3 and IF JNL NO = "X"
  do CONTINUE
4 and MOVE JNL NO TRANS JNL NO
5 and MOVE 0 BATCH TOTAL
6 and LABEL
ADD TO BATCH
7 and TRANS ENTRY
8 and LABEL
BATCH TOTAL PRINT
9 and BATCH TOTAL PR
10 and CS
11 and LABEL
INCOMPLETE
12 and BATCH END OPTIONS-
13 and IF BE OPTION = 1
  do TRANS CORRECTIONS
14 and IF BE OPTION = 2
  do TRANS DELETIONS
15 and IF BE OPTION = 3
  do GO TO ADD TO BATCH
16 and IF BE OPTION = 4
  do GO TO DELETE BATCH
17 and IF BE OPTION = 5
  do BATCH PRINT
18 and IF BE OPTION = 6
  do GO TO CLOSE BATCH
19 and GO TO BATCH TOTAL PRINT
20 and LABEL

```

```

DELETE BATCH
21 and DLF
22 and DLF
23 and DLF MG 38 "DO YOU WANT TO DELETE THIS BATCH ?"
24 and INPUT ANSWER
25 and IF ANSWER ^ "Y"
  do GO TO INCOMPLETE
26 and BATCH DELETION
27 and MOVE BATCH START NO NEXT TRANS NO
28 and SAVE THIS COMPANY
29 and MOVE "YES" BATCH COMPLETED
30 and CONTINUE
31 and LABEL
CLOSE BATCH
32 and TRANS SAVE
33 and MOVE NEXT TRANS NO NEXT TRANS NO
34 and SAVE THIS COMPANY
35 and MOVE NEXT TRANS NO BATCH START NO
36 and MOVE "YES" BATCH COMPLETED
37 and DLF
38 and DLF

```

```

RECAP
JNL HDG is a verb,
1 Does MOVE HEAD SAVE HEAD
2 and TITLE PR
3 and PXP
4 and PFP 9
5 and PRINT 16 JNL DESC
6 and PSP 18
7 and PRINT TD MONTH NAME
8 and PRINT " 19"
9 and PRINT TD YEAR
10 and PLF MLC 3
11 and TRANS HDG
12 and MOVE SAVE HEAD HEAD

```

There are journals, see, these are batches that I enter. After each batch I would print out the journal and screen it, to make sure that something wasn't in the wrong account — sometimes your fingers slip, you know. In the general journal, I put any entries that I used to make corrections. Once I had the journals and the batch was done, I would make out the general ledger, which is here.

```

RECAP
JNL LIST PR is a verb
1 Does PAGE HDG "JNL LIST"
2 and BEGIN JOURNALS
3 and LABEL
LOOP
4 and GET NEXT JOURNALS
5 and PSP 20
6 and PXP
7 and PRINT REF
8 and PSP 2
9 and PRINT "-"
10 and PSP 2
11 and PRINT JNL DESC
12 and PLF MLC 2
13 and GO TO LOOP

```

DBA 'S
EMPLOYEE PAYROLL RECORD
AUGUST 1995

LOUIS

41 SUNNYVALE, CA 94087
S.S. 32-0878 PAY RATE: \$6.50/HR AS OF: 3.02.95 DEDUCTIONS: S-1
DATE HIRED: 3.02.95

TRN	PRL DATE	CK. NO.	GROSS PAY	F. I. T.	F. I. C. A.	M. C.	S. I. T.	S. D. I.	NET PAY
4326	8.15.95	3000	\$193.91	\$0.00	-\$12.02	-\$2.81	\$0.00	-\$1.94	\$177.14
4338	8.31.95	3151	\$325.00	-\$33.00	-\$20.15	-\$4.71	-\$2.60	-\$3.25	\$261.29
⑥ MONTH TOTALS			\$518.91	-\$33.00	-\$32.17	-\$7.52	-\$2.60	-\$5.19	\$438.43
QUARTER TOTALS			\$518.91	-\$33.00	-\$32.17	-\$7.52	-\$2.60	-\$5.19	\$438.43
YEAR TOTALS			\$518.91	-\$33.00	-\$32.17	-\$7.52	-\$2.60	-\$5.19	\$438.43

WOLF: They claimed, for instance, that I could not have a total for that week, for the month, for the quarter, and the year print out right then and there. And originally it looked to me as if I couldn't — time and again I was running into a dead end, and I couldn't get this printed out from the file. Then I finally discovered what was wrong with it! You see these references here, the transaction numbers, they were in the general file, and I was working out of that file. And I would say, go in the payroll file, then take the next one, do this; take the next one, do this; take the next one, do this — but at the last reference in the file [the routine] would jump out. It was finished. And I couldn't branch back into the payroll program to print out these titles.

KIP: It was a question of recapturing the numbers associated with those transactions.

WOLF: Yes, but right then! [Laughs.] But I realized, first of all, that I had transaction numbers up to 5 digits built in — and I could even have enlarged that if I needed to. Then you see that on this sheet, my transaction numbers are 11,400-something, and that was in June. My transaction numbers would never go beyond five digits, almost certainly; so I put [transaction] number 99,999 at the end of the file. Then, when I went into the file, I would say "Get next transaction; if it doesn't equal 99,999 do this," over and over again, and then "If it equals 99,999, go into this program and print it out."

KIP: Because you knew that 99,999 was the end of the file —

WOLF: — and that it would never be a transaction number, yes.

KIP: There's your subroutine, Edwin.

```

RECAP
PRL PRINT is a verb,
1 Does DLF
2 and DLF MG DS 18
3 and DISPLAY "No. of copies
4 and INPUT COPY COUNT
5 and LABEL
    
```

```

REPEAT
6 and PAGE HDG "PRL"
7 and BEGIN PRL TRANS
8 and GET NEXT PRL TRANS
9 and MOVE PRL DATE WEEK
10 and MOVE PRL DATE WEEK PR
11 and PRL WK TOTALS 0
12 and PRL WK TOTALS CALC
13 and PRL TRANS PR
14 and LABEL
LOOP
15 and GET NEXT PRL TRANS
16 and IF PRL DATE ^ WEEK
do GO TO WEEK TOTALS
⑤ 17 and IF REF = '99999'
do GO TO END OF MONTH ⑥
18 and PRL WK TOTALS CALC
19 and PRL TRANS PR
20 and GO TO LOOP
21 and LABEL
WEEK TOTALS
22 and MOVE PRL DATE WEEK
23 and MOVE REF STANDBY
24 and PRL WK TOTALS PR
25 and PRL WK TOTALS 0
26 and MOVE WEEK WEEK PR
27 and BEGIN STANDBY PRL TRF
28 and GO TO LOOP
29 and LABEL
END OF MONTH
30 and PLF SLC NPG
31 and PRL TOTALS CALC
32 and PRL TOTALS PR
33 and SUBTRACT 1 COPY COUNT
34 and MOVE DIF COPY COUNT
35 and IF COPY COUNT <
do CONTINUE
36 and GO TO REPEAT
    
```

GENERAL LEDGER

PERIOD FROM JANUARY 1, 1995 TO AUGUST 31, 1995

ACCOUNT NO.	PARTICULARS	PRIOR BALANCE	CURRENT MONTH	BALANCE
				7
4205	ZCK CK. # 3161	8.31.95	BANK OF AMERICA	-13,053.00
4209	ZCK CK. # 3162	8.31.95	EMPLOYMENT DEVELOPMENT DEPT.	-5410.00
4212	ZCK CK. # 3163	8.31.95	REFUND	-30.50
4215	ZCF CK. # 3164	8.31.95	P.	-1,174.00
4217	ZCK CK. # 3165	8.31.95	& CO., INC.	-2,588.26
4219	ZCK CK. # 3166	8.31.95		-4,952.54
4222	ZCK CK. # 3167	8.31.95	MIKE I	-345.00
4224	ZCK CK. # 3168	8.31.95	- WEEKLY DISCOUNTS	-46.54

BANK RECONCILIATION

AUGUST 1995

BANK OF AMERICA #9275-03481

TRN	JNL NO.	DATE	REFERENCE	PARTICULARS	AMOUNT	
				LEDGER BALANCE	\$6,199.20	\$6,199.20 *
				OUTSTANDING ITEMS		
3980	15J	8.29.95	DAILY REPORT	DEPOSIT	-1,336.64	
					\$1,174.00	
					\$3,000.00	
					\$90.00	

HILARY: What about the bank reconciliation? Would this program allow you to clear the checks when the bank statement came in, and get your outstanding check list?

WOLF: Yes, and this is the program for a selection — it's nearly English. For bank first reconciliation, I would have the selection on the screen; if I wanted BR with an X — cleared — it would CONTINUE and jump out of the program. If not, number one is from bank one to bank one transactions, which just meant that it remained uncleared. Number two would go to transaction deletion. Number three would let me make an adjustment, number four would print it.

HILARY: So that's how you get your clear, clear with error, error adjustment.

```

RECAP
BANK I REC is a verb,
1 Does BANK I SELECT
2 and IF BR I = "X"
  do CONTINUE
3 and IF BR I = 1
  do BANK I TO BANK I TRANS
4 and IF BR I = 2
  do BANK I TRANS DL
5 and IF BR I = 3
  do BANK I ADJ
6 and IF BR I = 4
  do BANK I PRINT
7 and REPEAT
    
```

4224 ZCK 8.31.95 CK. # 3168
BANK BALANCE

\$1,174.00
\$2,588.26
\$4,952.54
\$345.00
\$46.54
\$26,083.00

MW: I start with [the screen] called Menu. First of all, I can change the date, if that was needed. If I was working in the same month, the date was in there already. And then I would go in and start with a batch that I had.

HLC: So you put in a batch of invoices?

MW: So I put a batch in. Every line that went in got printed out, so that if I needed to check back, I would see anything wrong on the printout. Then, when the batch was completed, the program took over and re-arranged all the accounts and printed out by accounts into the journal.

HLC: So if you had a stack of purchase orders, payables, you could put in the name of the company that you are buying from —

MW: I have five pieces of information that I put in: the account number, the date, the reference to the document, and then the description — vendor's name or whatever — and the amount.

HLC: And then will your program write the check to the vendor also?

MW: No, you see, my programs were made for once-a-month processing for the small company. My customers did the daily work themselves; in later years some of them even got computers to do their daily chores. I did payroll for just one company, but I didn't even bother writing a program because it was so small — I calculated the payroll manually, they processed on the computer from the time cards, and I printed out the payroll and the employee details. I made an exception for this one company only because it was so small -- about ten people. It didn't pay to computerize the initial calculations. But I stored the payrolls and the employee individual information all on the computer [post facto].

KC: Now, how long did it take you to write all the modules of this system?

MW: I think I worked about two months on the very first basic system, or maybe a little more. That [handled] the smallest account that I had. When that functioned, I tackled the next bigger account. To the very basics, I added whatever I

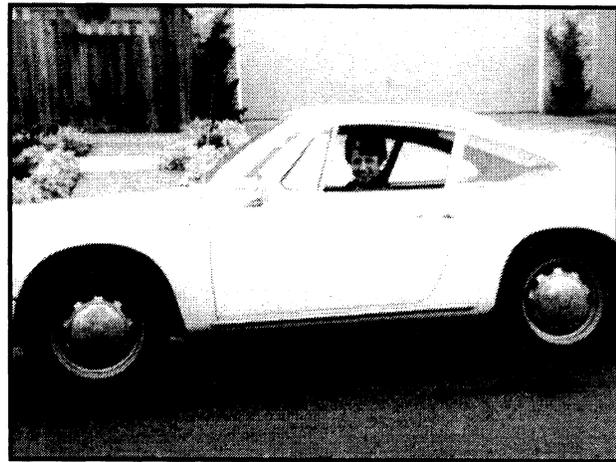
MW: I'm the kind of person who hangs onto things. Everything I have around me is old. I just sold my other car, a Porsche I drove for 29 years [with] the same motor, and put 300,000 miles on it. For the first, about, ten years it was the second car, so it never went on big trips.

HLC: Which model?

MW: It was a 912; actually, my 912 was the first one in California. When I was still married, the Porsche manager for the West Coast used to live across the street; when we went to Europe he arranged for us to pick one up at the factory in Stuttgart. But the 912 wasn't shipped to California yet, so when mine came in, nobody here had seen one. And I had trouble, because anywhere I parked I had ten or fifteen people standing around it. But cops were the worst problem! I don't know where they came from, but they were right behind me. And that's unnerving when you have a cop behind you all the time.

EVEK: A lot of protection for us!

MW: A guy called Rolf took care of it from day one until I sold it, and we overhauled the engine just once, and when I had another 120,000 miles on it [after that] he said "Well, it's still good for another 100,000 miles." The motor was okay, but of course all the rubber [gasketing] was getting old....



Maiga and her Porsche 912.



Removing the disk pack.

MW: I wouldn't sell the program. What he was willing to pay for the program probably wasn't very much, and there were no royalties, so why should I bother? I felt that [the program] was something different, and that I could give my customers something better. If everybody else started to have that, it wouldn't make any sense.

KC: Did you ever try to interest Logical Machines in the programming you had done, that they claimed couldn't be done?

MW: It wasn't the company -- it was one of the engineers from Lomac who had gone on his own. He had taken an ADAM or two and opened up an office where people could rent time on them. And when I said "I got the payroll the way I want it," and showed him the print-out, he asked me for the program. I said "Oh, no way." He said, "Well, my clients could use it," and my answer was "Then send them to me."

KC: And did he send them to you, or did he buy the program?

needed for the next account. Actually, I don't see any big deal about it. I had the machine do what I would do manually, followed my manual procedures exactly and just programmed them in there.

KC: But instead of working module by module, you worked company by company, starting with the smallest company?

MW: Right. So I got as far as printing out journals, and printing out the general ledger, but I did not work on receivables and payables and the payroll until the last. When I started to work on [those modules,] the people from LOMAC told me that the kind of payroll printout I wanted couldn't be done. I said to myself, there's got to be a way.

I struggled for some time. Then I discovered the reason that they said it couldn't be done, and I started to work around it. Then the payroll and payables became very simple for me.

KC: Now how long was it from the time you wrote the system for your smallest company, to the time you felt that your program was fully realized and capable of handling anything you

threw at it?

MW: About a year. I'm almost positive, not more than that.

KC: And I believe you mentioned earlier that, for example, by the time your system was finished every transaction was validated on the fly, and the system absolutely didn't tolerate rounding errors. It only took a year from the beginning to get it to that kind of sophistication?

MW: I had to rush — I needed that machine to work for me.

SERVICE WITH A SMILE

HLC: How quickly did people get their reports? What did they get in their reports?

MW: Depending on when they brought their daily paperwork in, the turnaround would be about three days. I screened [the data] after it was done and proofed it myself. This was for ten customers.

KC: So you were giving 3-day turnaround of reports for all the stuff to ten people? That was a lot of work.

MW: It was! But that's why it paid off that [computer] fast.

These are my copies, you see.

HLC: So you always had hard copy of everything, and didn't keep it on the computer.

MW: No. [It would be] deleted out of the computer. Otherwise there was too much garbage in there.

HLC: What about the paperless office? [Laughter.]

MW: I found that if I had too much information in there, it would slow down the processing. And I needed the time. My customers had no access to my computer [data,] so they had to look in their printouts anyway.

HLC: So, from being a manual bookkeeper, you went onto a computer, and suddenly something would take too much time because it took a minute, or five minutes. So rather than keep a lot of data on your disks, like all these month-to-date totals through the year —

MW: I would take time to transfer the information out of current work into a file for storage, because the computer had to work, and meanwhile I couldn't do anything else. Many of my companies had a lot of entries, between 500 and 1,500 for the month. So it was worth the time to transfer from the original file into storage.

HLC: How long would the computer take to close the month?

MW: I never timed it. This company would take the longest because they have the most entries. For the others, it depended on [the number of] entries.

HLC: But how long would it take [the computer to post the batch]? Was it short enough that you sat at the console while it did it, or would you go away from the computer for that time?

MW: Oh, I usually sat there. If I closed the month on this company — this was the largest that I had — I would let it run and come [across the room] to the desk. But for the others I would just sit there, because it wouldn't take long enough for me to start something else while it ran.

HLC: I notice you have a television over there. Would you watch TV?

MW: While I was posting I would. I would prepare the paperwork before I went to the computer.

Then I would know what entries should go in, and in general what transpired in that month, and the input was very mechanical. I would listen to public stations, some lecture or concert, while I was doing the input.

HLC: So then when you were finished, and it was sorting and posting the batches, and putting them into all the different accounts —

MW: No, the original entry automatically goes right into the account, and [the program] brings it from there whenever it's needed.

KC: That was how the transactions were validated, right? You checked the input against content, as soon as it went in.

MW: The input was checked for balancing as soon as it went in. And then when the journal was printed out — I don't have any journals [now] because I didn't print them for myself — I would check to make sure that paper clips were not under automotive, for example. So the journal gave me the totals, and each account would give me the detail. As the journal printed I would proofread it. The batches were automatically balanced as far as the numbers are concerned.

HLC: The debits equaled the credits.

MW: Then I would proofread the allocation to accounts when the journal was printed, and finally, when I printed the general ledger I would proofread it a second time. In case of an error I could go in and change it.

HLC: Oh, you could change it until you closed the month?

MW: For something that was wrong [in a prior month] I would make corrections through the general journal, but within the current month, I could change anything in the program except the transaction number — I could change the account number, the document reference, the date, the amount or the description, and correct misspellings anywhere. For instance, I had to hand-pencil something into my printed copy here — I put in 48 cents, I think this was a payroll correction. And then here, 56.79 in employee benefits, that went into the wrong account somehow.

HLC: So then you would change it by hand, key in the corrections —

MW: And print out a clean copy for the customer. But I penciled it in for myself when I found something wrong.

HLC: Then after you closed the month, if you had put this company in concrete pouring sales and the customer said, "No, no, no, we didn't do concrete pouring for them; we did drywall," — then would you reclassify the transaction through a journal entry?

MW: Yeah, I would change [the entry] from one account to the other through the journal, but the current entries could be changed anytime.

WORD OF MOUTH

HLC: When you were getting new business, did you advertise that you had a computerized system?

MW: I never advertised for business. My new business came through recommendation from my other clients.

HLC: Were new clients interested in the fact that your system was computerized?

MW: Oh, yes. After they saw what I could provide for them, they all wanted it right away.

HLC: What reports and financial statements could you give them?

MW: Whatever the company wanted, let's put it that way.

HLC: You had the ability to change the format?

MW: Most of the time I just changed the programs. I would get my new customers at the start of a small business, and usually they would have some problem with their bookkeeping. There was no point in overloading them with reports they couldn't use, I just gave them whatever was necessary. Then, as their company grew, I would "grow" the program to match. So I had every customer on an individual program, custom-programmed for them.

HLC: And you would give them a balance sheet and an income statement?

MW: They got all the printouts from me.

HLC: Now, you were using this system [from 1977] until October 1995?

MW: I retired as of October 31st, yes.

HLC: So they can't download their information from this into QuickBooks, or something. They have to just start over.

MW: They have to start over on their own systems because this doesn't translate into anything.

KC: You've said that any time you found an apparent disk drive error, which only happened lately, you traced it to a board on the computer. Did you ever lose any data off the hard disks?

MW: No, I haven't, not as a defect. When I started, Gary had me build in all kinds of things I called "idiot stoppers," that would ask me whether or not I wanted something. I took Gary's advice and put them in, but they made the processing too slow. Once I was on my own I took them all out. Well, at one point, I had the choice on my menu either to delete one transaction or to delete the whole batch; I pushed the wrong button and two hours of work went down the drain — I could see it go. After that I put an "idiot stop" in that place that said "Do you really want to delete this batch?"

HLC: So you knew, when it came up, that it —

MW: That it was a batch instead of one transaction. I also moved the selections further away, one from the other. Originally they were 2 and 3, and I pushed 3 and I should have pushed 2. So I moved the selection to delete the batch all the way to the bottom. When I wanted to delete just one transaction I wouldn't hit the wrong button.

SUPPORT

HLC: When the computer needed fixing, were you able to do a lot of that over the telephone?

MW: As a matter of fact, just in the last six months the battery went, and Raul [the current support tech] said "Well, since you plan to stop working and that battery's very expensive, we can fix it." And so we bypassed the battery with a little piece of wood, a piece of Chinese chopstick, and then I had it plugged into one of those outlets where I could turn it on and off. With the old battery I couldn't turn it on with a key, I needed a [new] battery. But when Raul told me over the phone to put the piece of chopstick in there, and I bypassed the battery so it worked.

HLC: So this has an ignition key?

MW: This particular [terminal] hasn't got it. We had to change my [CRT] tube in the last week that I worked. Raul came out and took the whole [terminal] from the spare computer and put it on this. He explained to me that the tube was supposed to last seven years, and this was the eighteenth year.

HLC: You got your money's worth.

MW: And so he said, we'll just switch it. The inside [of the terminal] is now from the spare computer, and I couldn't find the key. Since it would have to be turned on with a switch from the line voltage, instead of with the key and the battery, the key was not important. He felt there was no sense trying to chase down a battery in the last week. It was crazy.

HLC: And the second [ADAM] that you got for parts, how much did that cost?

MW: Seven hundred dollars.

HLC: Instead of \$32,000.

MW: Well, it was a used machine and Raul just got it, I would say, two years ago. He asked me \$700 and I said, okay.

EVEK [pulling boards and scrutinizing them:] Initially these looked almost like DEC PDP boards, but the backplane doesn't look right, it looks more like an S-100 type backplane.

KC: 16 Kwords.

EVEK: 16 Kwords?

KC: 16K 16-bit words.

MW: I know nothing about it. If Raul told me over the phone "Do this, and push that button, and look at that," then I could do it. But otherwise I have no idea what it is. Right now it has original boards in there, some boards from the second computer, and some of Raul's boards. Last time he came in, he said "Now we're not going to mess

HLC: Once you paid for the machine, how much business expense did you have?

MW: Not much; what I needed was stuff like paper and ribbons and paper clips. It wasn't much of an expense. Now, for a number of years I also had a china shop.

HLC: Did you keep your bookkeeping business and your china shop business going at the same time?



MW: In 1985 I decided that, as long as I could still work a little harder than necessary, I would open the china shop for my retirement. I felt that as I grew older [the bookkeeping] was taking quite a bit out of me -- it required a lot of concentration about the things that I had to remember or else. And then no sooner was there a deadline than my customer would come in two days beforehand and give me five days' work.

HLC: Right, and you'd work all night.

MW: Oh, yeah. So I thought, well, I can always run a china shop, that's no problem, so I bought into a little shop and I was running both businesses. But from '85 to '88 the shopping center that this shop was in went down the tubes; the nice shops went out and Payless and [some other discounters] moved in, and I was too fancy, I had only good quality china and crystal and stuff. So this shopping center wasn't doing much for me, and in April [1989] I moved over to Saratoga. In October we had the [Loma Prieta] earthquake and -- I've got a lot of pictures to show. [Laughs.]

HLC: And you got insurance, right?

MW: Yes, but silly me, being in the business I didn't think rationally. When I got the insurance money was when I should have packed up and gone home. Instead I thought "Well, gee, so many people have broken dishes now that it'll be a godsend to them if I can supply them with good dishes." So I took all the money, refurbished my store, and within two weeks I was back in business. The ladies came in and said "I'm not going to buy anything breakable any more." That was the attitude. And right after that the economy went down, my

around," because I had three more days' work and the thing conked out on me.

EVEK: Just had to get things finished up to give it back.

MW: We just patched it up so that I could work on it to finish up. At least it lasted long enough.

EVEK: It's still astonishing to me that you have \$32,000.00 in that, that's a lot of money.

MW: But if you earn \$40,000 to 45,000 a year, in a couple of years it's [amortized]. It paid for itself.

KC: Handsomely, I would say.

EVEK: And then when you think — I get a new computer about every two years.

KC: Right, you get a new computer every two years, but there are computers and computers; and this goes right to the heart of the old debate that always ends with the line "It isn't the cost of the

hardware, it's the cost of the software." The major cost here was actually the development time and the debugging and refinement time of the software.

MW: What I put in there was what I wanted and needed. Now, truly, once the system was working and I was working fluently right through, I found a lot of other things that I still could refine — make an even shorter verb or something — but it was working, so who cared? I could see areas where I could make improvements, but it was giving me the answers anyway, so I didn't bother spending the time. By then I had all the customers and I was too busy to tinker around. Some customers would ask me "Oh, do you think that we could get some percentages on this statement," and sure I could get them, so away we went. But if people didn't want percentages, I didn't offer them.

KC: Right, you don't give people features unless they ask for them.

lease came up, they wanted more money, and I wasn't going to work just for the rental, so I packed up and moved everything here. I still have some \$15,000 in china samples sitting in the garage.

HLC: Hope there's not another earthquake.

MW: Well, what are you going to do? I had a big sale and fairly well sold out the regular merchandise, but I couldn't get rid of the samples. How can you sell one place setting of each pattern?

KC: I have to say at this point that you may be the only person I know who went into retail for relaxation.

MW: I'll tell you, for me it was very simple -- the paperwork was a cinch. I could devote my time to the merchandise and to the customers.

HLC: Did you yourself work in the store, or did you have employees?

MW: Just myself.

HLC: Just yourself. How many hours a day did you have it open?

MW: It was open from ten to six, and then I'd come home and work [on bookkeeping] until about two o'clock in the morning. At ten o'clock I'd be back at the store. I worked 16 hours a day, 7 days a week for 7 years.

KC: Hm. Okay.

MW: Well, actually I didn't work 16 hours towards the end. Some of the customers retired, and I didn't take any new customers, so gradually I went from 10 down to 4.

HLC: In the bookkeeping.

MW: And then after I closed the store I kept the four and didn't add any more. See, I'm not a spring chicken anymore, I can't keep up. An old lady is not a bullet train. You have to make stops here and there.

MW: Now, sometimes I would say "You need this for your own good," or "I think you need detail of this or that." But there are all kinds of customers. I had some who were only interested in the bottom line, just how much profit was made, that's all they cared for. They couldn't care less how you got it. Then I had customers who would look into it but were not very interested, and then I had some who wanted to know everything about the general ledger — how to read it, how to find things that they wanted to look up. So there are all kinds of people.

I charged them according to a system figured on the base work that I needed. Like the one account I showed you had a lot of entries, so my base charge would be high. For those who had only 100 or 150 entries, my base charge would be low. In addition I charged a dollar per thousand on gross income, so if they started as a small company and kept going up, my fee automatically would go up [with their gross]. I also charged \$5 per person on the payroll every month, and when it was time for the quarterly reports, I didn't have to charge extra — that was prepaid already. If the company grew, my fee automatically went up, and if they had no income at all, I still had the base.

HLC: That's a good way to charge.

MW: It worked beautifully, and I never had to ask for an increase.

KC: This was a complete custom system. Who would sacrifice the hardware platform that it ran on?

MW: I'm sure that [computer] system is a lot more capable than what I used it for. If I needed more information [in the ledgers] then I could have put in more [in the program] but I didn't need it. I only put in what I needed.

KC: When I say the capabilities of the system, I'm not talking about the hardware specifically, I'm talking about the capabilities of the accounting software.

MW: Oh, okay. I thought the computer will do anything you just tell it to do. I don't think there is any machine that won't do what you want. If you know what you want, you can put it in there.

Quick Take: RAMAC 40th ANNIVERSARY!

That's right....the fortieth birthday of RAMAC, sometimes known as the IBM 305, the world's first hard disk. The fact deserves some thought — since in that time a CPU has turned from rack on rack of firebottles, to TTL, to a single IC; memory shrank from ferrite core arrays to daughterboards crammed with RAM chips; I/O was a Teletype then and a color flatscreen now. Yet we're only *beginning* to think about what *might* replace the hard disk, a device as spectacular now as it was in 1956. Three cheers for Rey Johnson and the fine engineers of IBM San Jose — and many happy returns.

(If you don't know the stunning story of RAMAC, you might want to order ENGINE 1.4 and 2.2, which are still available from our back issues department at US\$6 each. Overseas customers please add US\$3 postage.)

Quick Take: GETTING FRAMED IN 3D

Did you happen to notice the nicely framed and glassed Apple One that sold for \$22,000 at the 1996 Computer Bowl? Would you like to frame your own favorite singleboard — or the circuit that made your reputation? Problem is that an ordinary downtown framing shop, even a competent one, will sometimes back off when queried about a box frame. But the ever-vigilant CHAC has found a framer who won't flinch. Contact:

Frame-o-Rama
210 Hamilton Avenue
Palo Alto CA 94301 USA
+1 415-321-3939

and ask to speak to Victoria Miller....who says she truly enjoys framing an occasional computer. The end result is gorgeous, too.

THE MAC AND ME: 15 Years of Life with the Macintosh (Part 2)

by Jef Raskin

ANTI-MICRO ATTITUDES

The computer industry in the middle 70's tended to ignore or minimize the microcomputers that I saw as the future of computing. (The term "personal computer" was to come later.) Nonetheless, I was invited to chair the National Computer Conference session on documentation in 1979 — but this was mostly on the basis of my presence in the large-computer world.

At first, those who asked to exhibit microcomputers were turned down. By 1978 they were given a room in the basement. A few of my friends at the large computer companies asked me why I was throwing away my career by working for a microcomputer company.

At one of the National Computer Conferences I was on a panel where I was expected to uphold the proposition that microcomputers were useful. Many mainframers thought and said that micros were - and would remain - toys. We each gave our little talks, but I didn't score until the discussion session.

To show the superiority of large computers, one of the speakers challenged me to some "benchmarks." The exchange went something like this:

"Anything your little Apple can do, my mainframe can do, and do it better," he boasted. "For one thing, microcomputers don't have the speed of a mainframe!"

"OK," I replied, "name your speed benchmark."

"Invert a 100 by 100 matrix! It will take me about 40 seconds."

"You win," I conceded. "It would take my machine hours to do it."

The audience gave a bit of applause for the mainframe.

Then it was my turn; "We both have to run across the hall. The person getting to the other side first, carrying his computer, wins."

There was laughter as people pictured him trying to pick up and run with his mainframe, larger and heavier than a refrigerator, and then there was a solid round of applause as I raised my Apple II with one hand.

"For my next benchmark, let's discuss power," he said. "Have each of our machines create an index to a thousand page book."

I had to concede. My computer couldn't even hold that much text. This admission got a few guffaws from the audience.

Then I proposed my second benchmark: "You take \$100 out of your salary every month and I'll take \$100 out of mine. The person who can pay for his computer first wins."

There was a lot of laughter and applause. "But," argued my opponent, "that's not computer power!"

"A computer," I answered, "has no power at all if you can't afford it." From the audience reaction, it was clear I had won the debate.

MACINTOSH PROJECT PRELIMINARIES

Early in 1979, probably in March, I talked with [Mike] Markkula about my idea for a new computer. He had had an idea for a \$500 game machine, which he called "Annie."

I thought that a game machine, although a good idea, was not something that I'd feel comfortable doing. So I counter-proposed a general-purpose, low-cost computer based on my own ideas - and dreams - for an interface. Markkula agreed to it.

I picked "Macintosh" as the name for my project, since Macs were my favorite apples. I changed the spelling because I wanted to avoid conflict with the name of an electronics manufacturer - an attempt that proved to be in vain.

Most of all, I didn't want to call the project "Annie," since I felt that the trend in the company to give new products feminine names was sexist — and if you had spoken to the namers you would agree.

Markkula's "Annie" project would, besides games, have allowed the user to program in BASIC. But it was not intended for business, and I thought any new product should be able to handle a much wider range of applications.

I also said that using a TV set or a third-party monitor was playing Russian roulette with one of the most important selling points of a system - how the screen looked.

With these wants and limitations in mind, Markkula sent me off to do design and cost studies. Working with my friends at Apple, notably Brian Howard, I came back with an absolute minimum selling price of \$1,000, far from Markkula's goal.

The machine I designed was based on the 6809 chip and had a 256 by 256 bit-mapped screen. I came up with a proportionally-spaced character set that would display 25 lines with an average of over 80 characters per line on the little display. (To put this into perspective, the Apple II displayed only 40 upper-case characters per line. The idea of proportional fonts on a display was then unknown at Apple, though commonplace at PARC.)

My choice of the 6809 was dictated by the tight price constraint imposed initially by Markkula. The better 68000, when it first became available a little later, was \$400 - if we bought it in quantity. That would have made the product have an introductory price of about \$3000.

My original concept was biased toward the inexpensive and memory-efficient. I noted that a 256 by 256 display could be addressed in exactly two bytes, making fast software easier to write - speed is of the essence in a good interface.

To convey one of the Macintosh design features to others in the company, I built an Apple II with a monitor incorporated into the lid. I used it at lectures and demos and it had great appeal wherever I demonstrated it.

(To this day I don't know why Markkula - to whom I pitched the idea the strongest - Jobs, and all the other people in management didn't use my idea in the II. The Apple II had a pop-off lid, and we could have sold a replacement lid with an angled CRT built in.) My very happy experience with this prototype settled it: the first Mac would have a built-in display.

FRICION WITH JOBS

While the company was thinking about manufacturing tens of thousands of computers a year (another unheard-of idea), I wrote an internal document called "Computers by the Millions." In

it I looked at questions of design, manufacturing, marketing, and general social and economic impact of computers in those quantities. Management found the paper valuable, and would not allow me to publish it for three years, to avoid letting the competition know what we were thinking. It was still years ahead of its time in 1982, when I published it in the ACM's SIGPC Bulletin (Vol. 5 No. 2).

Jobs, unaccountably, did not at all agree with my views of the future, nor with my distributing them internally at Apple, even though I was doing so at Markkula's request. By proposing new strategic ideas and products independently of Jobs, I began to get on his "wrong" side. By this time Jobs had begun to have people who were "in" and those who were "out;" if you were "in," everything you did was golden, if you were "out" everything you did was rotten. By the time Jobs had started NeXT this had become a major trait of his, according to Randall Stross's book, "Steve Jobs & the NeXT Big Thing." My take on this book, and its view of Jobs, appeared in 1994 as "Hubris of a heavy-weight" (*IEEE Spectrum*, July 1994, pp. 8-9).

But as I began work on the Mac, I didn't recognize the Jobs phenomenon. Thinking I was still "in," I kept on trying to get Jobs to go see what PARC was doing; since I was actually "out," he resisted the idea strongly.

I, of course, remained oblivious to what was going on. I thought that he would turn around as soon as he saw the quality of what I was doing. Besides, we had been friends, and our disagreements were purely technical.

PASCAL

Early in 1979, I tried very hard to convince the company that we should move away from using BASIC and assembler as our main languages for applications and system software. After presenting the case for and against a number of major computer languages, from FORTRAN to APL, I argued that we should base our work on Pascal. I hired a clever and inventive ex-student of mine, Bill Atkinson, who implemented a Pascal developed under Ken Bowles at UCSD. They had it running on the 6502 processor, the same processor used in the Apple II, and Atkinson suggested porting it to our product.

In the process, Bill had to write graphics routines, an experience that proved extraordinarily valuable for Apple. Many in the company had rejected PASCAL as impossible to put on an Apple II, contradicting several technical memos I had written showing how it could be done.

As Atkinson later said, "We had a bunch of self-trained amateurs who didn't really understand modern software development. The system software team actively resented a new language. Once we had it up enough to demonstrate the word processor, and Markkula saw that, it was clear sailing."

I supported Bill's implementation, and then wrote a PASCAL manual with Brian Howard. Pascal, as I had predicted, allowed us to hire more professional programmers, and later became the main development language for "Lisa" and the Mac. At the time, I personally paid a license fee to UCSD so that Apple could use their Pascal system. Apple never reimbursed me, since Jobs insisted that Apple didn't need and would never use Pascal. Almost all Mac and Lisa software was written in Pascal derived from UCSD. I remain amused by the thought that in some vague sense, it was all owned by me.

THE MAC BECOMES OFFICIAL

By September 1979, Mike Markkula had - over Steve Jobs's objections - approved the Macintosh project. But by going around Jobs I had unknowingly set up a dynamic that made the project far more difficult politically than I could have anticipated.

From the first, Jobs opposed it, calling the Macintosh the "dumbest idea" he'd ever heard of. He would often recite a list of imagined advantages that the Lisa project had over the Mac and put obstacles in the way of my obtaining staff or supplies. His interference eventually became so overt that Mike Scott had me move the entire Mac project to some buildings behind a Texaco gas station across De Anza Boulevard, so that we would be able to develop the Mac in peace. Since we were on the second floor, we called it "Texaco Towers." Later, when Jobs took over the project, he put up a pirate flag and claimed that he moved the Mac out of Apple headquarters so that it would remain pure and uninfluenced by the stodgy company engi-

neers. To me, the pirate flag really indicated a pirate within: as I see it, Jobs took over the project by fiat and lies, and was nearly successful in stealing the credit for having originated it as well.

From the beginning, to keep the project on track, and so that we would not lose good ideas (and the reasons for abandoning others) in the press of development, I created a document numbering system and put the collected documents in the "Book of Macintosh," which grew to some 400 pages. I wrote most of the book, since I liked to write - and was the fastest typist in the group - but the ideas were generated by everybody, and everybody got credit in the text.

Here's one example of the standardized heading format:

```
MACINTOSH PROJECT DOCUMENT 18
VERSION 0 DATE: 20 OCTOBER 1979 TITLE:
DELIMITING STRINGS. AUTHOR: JEF
RASKIN.
```

I asked all the participants to explain the reasons for their conclusions, their right turns and wrong turns, as we went along. I wrote most of the documents late at night at home; we were too busy during the day to get around to it.

I believe Jobs's opposition was partly due to his not understanding what I was trying to accomplish, though at the time I incorrectly thought of him as the supportive friend he had been for so long. For example, when I insisted on bit-mapping and square dots, he would retort that Woz had put a character generator in the Apple II and it didn't have square dots and its sales were paying my salary.

THE FAMOUS ANACHRONISM

It has been often said in the computer and general press that the Mac was a straightforward copy of the work done at PARC. It was not, and the idea does a disservice to the hundreds of people at Apple who developed the hardware, software, marketing, and interface concepts. This erroneous belief turns what was a significant intellectual debt into the appearance of moral bankruptcy. I can't blame people for making the mental leap from hearing that the Mac resembled - and was partially inspired by - the interfaces at PARC to guessing that it was largely "stolen." But at the same time I cannot forgive those who write on the subject (and

make the same claim) for not doing their homework. Most of them never saw or used an Alto, a Dorado, or a Star — the systems developed at PARC. They simply assume that the earlier systems were much the same as today's Macs and Windows machines.

Then there is that apocryphal story about Steve Jobs visiting PARC, having an "Aha!" experience and coming back to Apple in full cry to create the Macintosh project.

Well, he did go, he did see, and he did come back enthused, but the Macintosh project was well under way at that time, having been officially started months earlier. The trip was set up to convince him of the value of the Macintosh project. I'm not sure how the story got reversed, but I later learned that Apple's PR department repeatedly told the false tale to anybody who asked.

Nearly a decade after the introduction of the Macintosh, Xerox took Apple to court over the issue. I was briefly invited to be an expert witness, not by Apple - as I might have expected - but by Xerox. The Xerox attorneys soon learned that the main thing I could testify to was the originality of the work done on the Macintosh. (The Lisa group did do what I consider some shameless copying of the Xerox Star, down to the names for some individual fonts, but that is a different story arising from the fact that a lot of key people on the Lisa project had been hired from Xerox, something that was not true of people in the Mac group.)

The case did give me a chance to use a Star and an original Lisa, each for the first time, an experience that taught me how much further the Mac was from its predecessors than I had remembered.

INTERFACE INNOVATION

One of the substantive differences in the "look and feel" of the Mac interface was the one-button mouse. The one-button paradigm has become so pervasive that many applications for IBM-compatibles ignore the second button that clutters most IBM-compatible mice; the third button that was part of the Engelbart and PARC mice has also disappeared almost completely from popular use. My own difficulties with the three-button mouse - and watching other people have trouble learning it - led me to rethink the design.

With one button, I reasoned, you could not get confused about which to use. It took a while, but I was able to find methods that in every case required the same or fewer operations than those required by the PARC system; it was faster, easier to learn and use, and it was far less "modal."

Of the methods I invented, the most fundamental was the idea of pressing and holding a button while dragging, and using the release of the button to indicate that the operation was complete. This differed from the method — used at PARC and dating back to the work of [Ivan] Sutherland — of click, drag, and click again.

When Larry Tesler came from PARC to join Apple he was naturally resistant to the one-button mouse. Larry was comfortable with the three-button implementation and had long touted its advantages over non-mouse systems. It took considerable effort to convince him, point by point, that my solution was not only workable (which he and others doubted at first) but in fact superior.

In any case, the interface we developed was a distinct and new creation, though it shared many elements with and owed a very real debt to what had been done at PARC. A major part of that debt, of course, is that I was able to use PARC's work as a living demonstration of a highly evolved graphical interface.

The one-button mouse was not the only major difference between the Mac and the systems at PARC. Another interface improvement that made the Mac feel so much easier to use was the way a user selected something or engaged a menu. At PARC, menus were relatively static lists of limited length that the user could summon and dismiss. Bill Atkinson — later to become an Apple Fellow — proposed that we instead extend my method of selection and drawing so that just the title of a menu would be shown, but when you pointed to it, clicked and held down the mouse button, the menu would appear! Then you would release the mouse button when the cursor was pointing at the desired item. This made menus appear when you needed them and disappear without apparent effort. Furthermore, as we both pointed out, having the menus at an edge of the screen and having the cursor position confined at the edge meant that you had to point accurately in only one dimension, which made the menus easier to use. The design of Microsoft's Windows and similar

interfaces does not have this useful “pin to the edge” idea.

Atkinson was led, by analogy with my point and drag methods, to pulldown menus that you can drag across to your desired item. Probably because it worked much as typewriter SHIFT keys do and as a pencil does — you put it down at the beginning of a line and lift it up at the end — my method of using a mouse has prevailed.

I extended this idea to drawing lines and to creating rectangles and other shapes by pointing and sweeping across the diagonal. My “hold and sweep” concept was then applied to making graphical selections. We created a rectangle that surrounds or touches the items to be selected while the button was held. The methods I devised are now so universal that some people who worked on the earlier systems have forgotten how they worked. They tend to “remember” them working as the Mac does now. What I remember is the effort it took to convince my fellow engineers that what I was proposing was better.

I suggested that Apple patent the one-button mouse and the new way of using such a pointing device, but Jobs nixed the idea in favor of patenting Atkinson’s pull-down menus. Apple missed this opportunity simply because Jobs didn’t want my name to appear on any Apple patents (though I have about a dozen of my own). I was still “out.”

A more subtle difference between the Mac and the work at PARC is this: in the Mac you point to something and then tell the system what to do with it. It is the “noun-verb” paradigm that is now nearly universally recognized as desirable by interface designers. As Bill Buxton of PARC has reminded me, the Xerox products used a more complex noun-verb-noun method involving a bunch of function keys (like the current IBM compatibles).

To quote Buxton, “Both the concept and the operation were quite different... it is remarkable how few people who teach and talk about GUIs even seem to understand the differences to even this degree of subtlety.” Buxton and his colleagues also published research in the 1990s (on what they term “kinesthetic feedback”) that showed why my click-and-drag paradigm worked so well.

Another fundamental part of the Mac from the very beginning was the insistence that unifying software would be built in. Knowing the time con-

straints of the real world and the inherent laziness of all humans, I suspected that if we built in an interface, programmers writing applications would use it, grudgingly, for their first mock-up as it was much faster and easier than writing the interface themselves - standard practice in all products prior to the Mac.

I knew that writing a rule book would only antagonize the independent spirit of software developers, who are inherently entrepreneurial. They had to be tricked into using the Mac interface. I could depend on their time constraints, and the likelihood that our interface would be far superior to what they planned, to insure that the details enforced in the software prototype would appear in the final product.

It worked. When the Macintosh was released, users found that learning new applications on a Mac required far less effort than the same task on any competing system. This gave third-party software developers added incentive to do things in the Macintosh manner, and Mac users have reaped the benefits.

The success of the Mac led other companies to copy its interface, and one can now move without too much difficulty from the Mac to Windows, to Geoworks, to most workstations, and even to some mainframe front-ends without retraining and with barely a glance at a manual or help screens.

My unifying software originally was to be a graphics-and-text editor within which applications could run as additional commands (via menus), all input and output being through the interface designed for the editor. Later, the PARC desktop metaphor was adopted from the Lisa group, who had adapted it from the Xerox Alto and Star computers. The incredible work of the Mac software team designed and squeezed the necessary code into a “Toolbox” within a relatively small ROM (Read Only Memory) that we could afford to put into the product.

The interface concepts I wanted to implement required fundamental hardware changes. One example was the way the electronics of keyboards were designed, not in keyboard layout - which obviously affects the interface - but in the way the keyboard works at the chip level. Before the Mac, and excepting PARC which was at that time not a commercial manufacturer, the makers of commercial keyboards built each key to put out a signal

when pressed. By the middle 1970's a special "encoder" chip took a signal from a key and produced the code for the symbol that key represented. There were usually a few exceptions: the SHIFT key could be pressed and held and while other actions took place; the same was often true of other state-shifters such as the "control" key.

But these exceptions were built into the encoder chip; what I wanted was a keyboard whose keystates — whether any keys were up or down — would be "known" by the computer. By analogy with pianos and organs, which can use any combination of keys simultaneously to play what musicians call a *chord*, this was known as a chord keyboard. I had long believed that this was an essential step toward improved interfaces and when I first went to PARC I was delighted to learn that they had come to the same conclusion.

Burrell Smith, our hardware designer, participated avidly in these discussions, and often suggested ways in which hardware changes could help the interface, sometimes also proposing changes in software design that could simplify hardware requirements. In each case the interface requirements took precedence, but this was probably the first time a commercially successful computer was designed with hardware and software subservient to the issue of usability.

The Mac succeeded because the initial impetus for its creation came from a humanitarian impulse, rather than a hardware dream or a marketing study.

SELLING JOBS ON THE IDEAS

A popular description of Jobs is that he has a "reality distortion field." This phrase accurately described Jobs's ability to convince people that whatever he was saying at the time was inevitable. I'd seen him charm otherwise reasonable people into believing absolute nonsense.

Some of this is helpful when doing something new in the world, but - as I see it - Jobs lived at the center of this field and actually believed and acted not only on vision, but on the basis of his own falsehoods, sometimes with unpleasant consequences.

It seems to me that Steve Jobs was also mesmerized by the power of part of his key insight that had helped make the Apple II a success: 'make it look

attractive' became a guiding principle. He continued to confuse appearances and quality ever after; years later at NeXT, his first major expenditure was to hire decorators for the new office complex. This passion for appearances would have been an asset, or at worst immaterial, in someone who also understood the products; but Jobs often did not.

THE PASCAL POSTER

In 1979 he botched the design of a poster that summarized the structure of the Pascal programming language. Programmers found sets of diagrams created by the originator of the language, Niklaus Wirth, a handy reference. In writing the Pascal manual I had discovered several errors in Wirth's diagrams and also disclosed some simplifications. Diagrams in the manual reflected these corrections and improvements. I thought that it would be good advertising, as well as a real benefit to programmers, to put the entire set into a decorative poster. Color would serve as a key to link items of the same syntactic type, making relationships among language elements clearer.

Jobs thought it was a great idea, and promptly hired a prominent graphic artist, Kamifuji, to produce the poster. Jobs asked me for a copy of my diagrams so that the artist could estimate the project, telling me that once we had a quote I would work with the artist. But the next thing I knew, Jobs proudly came into my office with the finished work. Thousands had been printed.

The poster was very good looking, with bold colors on a jet-black background. But some of the diagrams were no longer correct and the colors had been chosen purely for esthetic effect, making the chart unnecessarily hard to use. I told Jobs that it was very pretty but wrong; he didn't care, and blissfully went on to something else. The posters were sent to stores as advertising posters, but they couldn't be shipped with the Pascal product as planned. It was a waste of time and money. This example is not significant in the history of Apple *per se*, but does say a lot about how Jobs thought.

MAC POLITICS

By the end of 1979 it was clear to many people that unless Jobs had a better understanding of what was being attempted on both Lisa and the Mac, he

would continue to inadvertently sabotage the former and be antagonistic to the latter.

My friend Bill Atkinson knew a great deal about what was going on with the Macintosh even though Jobs had officially forbidden him to work with members of the project. This meant that Bill had to keep his involvement with the Mac secret, lest he lose his "in" status, while he worked on the Lisa. At the time he was writing his meticulously crafted QuickDraw graphics system - then called LisaGraph - for Lisa (Apple knowingly used the name of the list-structured graphics system I designed for this central piece of software - without permission or compensation).

With Bill's connivance and the help of Tom Whitney — who had given me the title of "Manager of Advanced Systems" to correspond with my work on the Mac — and by keeping my name out of the picture, we at last managed to convince Jobs to visit PARC.

Jobs later said that after he went to PARC, he returned inspired, and launched the Lisa and then the Macintosh. This story, once promulgated by Apple's PR department and often repeated in books, articles, and even by the generally excellent PBS series on the history of the computer, is inaccurate, to say the least. As Atkinson put it in a phone call to me, "You were instrumental in getting Jobs to go to PARC, and that was central to getting his support for new interfaces." Jobs pointedly did not invite me on this visit, and excluded me from the conversations about it when he got back; it would have been very hard for him to have admitted that I had been right about the value of the work done at Xerox.

In general, I remained oblivious to the politics going on at Apple, and concentrated on the design of the Macintosh. This left almost no room in my life for anything else, except practicing the piano and occasionally getting out to fly a model plane. I bought a house a few blocks from Apple so I could bicycle in and back on a moment's notice. The Macintosh project was my life.

CONCEPT AND COSTING

What was the Mac concept like in the early days? We researched many possibilities. For example, we considered a bit-mapped LCD display which had a resolution of 256 X 26 (yes, twenty-six) and a cost

to us of about \$240. At our usual five-to-one ratio of parts cost to list price, that part alone would have been \$1200 at retail. A 256 X 256 or larger display with any technology other than the cathode ray tube (CRT) was then totally out of the question, since a CRT display cost between \$35 and \$50.

A drawing done by Brian Howard in 1980 shows a one-piece box with a built-in CRT, 5 1/4" drive, keyboard, and joystick. The joystick is in the same position occupied by the trackball in the later Mac Portable.

We also worked on a strain-gauge stick almost identical to the current IBM graphic input device. Embedded pointing devices have a long history at Apple; for example, in 1978 Woz came with the idea building a pair of orthogonal thumb wheels (one each for vertical and horizontal motion) under the Apple II keyboard. This was a response to my request that we build a pointing device into the box that could be operated without removing the hands from the keys. This seemingly obvious good idea reached fruition years later with the PowerBook series and was probably reinvented independently by the PowerBook group.

Graphic input was an essential element of the Macintosh from the first. I thought that the mouse in particular was a clumsy way of going about it - for one thing, it takes up too much desk space, and for another you have to find it anew each time you want to use it.

But Jobs was an adamant mouse-ist, (mainly, I think, because that's what PARC had), and until third party vendors supplied trackballs, the mouse was the only graphic input device available for the Mac.

TEAM BUILDING AND THE TOOLKIT

One of my basic concepts was of a software nucleus that would be built into ROM, and serve as a home port to the user, tossed about on the high and varied seas of application software. To write the software, I hired Bud Tribble, who had similar thoughts. He and the two other "B's," Brian Howard and Burrell Smith, were the first Macintosh team.

The Mac Toolkit was initially written by Tribble, who was in charge of Mac software; it was taken over by unstoppably hard-working UC computer

science dropout Andy Hertzfeld, Bill Atkinson, Bruce Horn (who, at fourteen, had been one of the usability testers of Smalltalk at PARC), and others.

Each member of the original “gang of four” came to the group through a different route. Bud Tribble was a medical student, a programmer and designer of genius who I had known at UCSD; he and Bill Atkinson had been good friends there. Atkinson pointed out the talents of a man working in repair, Burrell Smith, and after interviewing Smith, I hired him as head of hardware design. Brian Howard had been a friend of mine for years.

There were established hardware designers that I had tried to bring over to the Mac (and who wanted to work with me), but Jobs had forbidden them to join the project. Still, Smith proved a first-rate designer who was open to thinking from a software and human-interface point of view, and he was a delight to work with.

I brought MIT anthropology student Joanna Hoffman on as our marketing person. Her major contribution to the Mac was to make sure that design decisions didn't preclude international sales; this concern was unusual in the then-parochial microcomputer industry. Thus the Mac had, from the first, the accents, special characters, and diacritical marks needed in languages other than English. We had come a long way from the philosophy that upper-case letters were all you needed. (Joanna also introduced me to my future wife.)

Steve Clark (another UCSD student I brought to Apple, and whose Olympic-level-kayaking sister Candi was later to marry Woz - as I've said, it's a small valley), and a few others formed the nucleus of a team easily the equal of the much larger and better-funded Lisa group. I hired some, such as programmers (and musicians) Gareth Loy and Bill Schottstaedt, from SAIL; another, the remarkable poet Bana Witt, had been a music student of mine when I taught at the San Francisco Community Music Center. She later married Bruce Tognazzini, another Apple employee who worked with me and was to write and lecture extensively about interface design. (Being a minister, I had the pleasure of conducting their nuptials.)

Donald Reed, the very image of a bookish intellectual in appearance and manner, worked with me closely on documentation. Of course we enjoyed the under-the-table help of Atkinson and others on

the Lisa team who believed in what I was trying to do, and the warm support of the late Tom Whitney, who had been hired to head engineering for all of Apple.

THE END OF THE BEGINNING

John Couch, a good manager and insightful computer scientist who was running the Lisa project, increasingly found Jobs a nuisance, and eventually managed to get him removed from the Lisa project. Jobs, at loose ends and hearing rave reports about the Macintosh, decided to have a hand in it.

Apple's top management helped shunt him to the Mac project to get him away from Lisa, which was seen as the company's hope for profitability in the 1980's.

Jobs's attempts to undermine the Mac project now took the form of destroying my credibility. One of the more blatant incidents was the “brown bag” lunch at which I was to describe the Macintosh project to the company at large. It is discussed in a confidential memo that I wrote to Mike Markkula to explain why, though I was seeking someone to manage the Macintosh project so I could concentrate on technical issues, I didn't want Jobs to be in charge. The memo specified, in detail and in my judgment, Jobs's many and egregious failings as a manager.

I had asked that the memo be kept secret, and Markkula agreed, though he said that he didn't think he could do anything to control Jobs. I believed this assurance and, thus, felt betrayed a few days later when Jobs called me in to his office to “discuss” the memo. I dimly recall Markkula saying something about having had to discuss it with Jobs. But Apple was a very open company, doors were left unlocked, and people wandered freely into one another's offices. Any of a number of people might have seen the memo and made a copy for Jobs, or he may have noticed it himself. Markkula thinks that something of this nature is what must have happened, and it might well be.

The memo reflected the running joke that the way to get Jobs to agree to something was to tell him about it, let him reject it, and wait a week; when he came running to tell you about “his new” idea, you'd exclaim, “Great, Steve, we'll do it right away!” In the memo I also made a prediction that was to prove exact: “Jobs was wrong on his Apple

III schedule, wrong on the Lisa schedule, wrong on the cost and price estimates, and he will be wrong on Macintosh. He is a prime example of a manager who takes the credit for his optimistic schedules and then blames the workers when deadlines are not met.”

The memo also related the incredible brown bag incident: “Jobs is often irresponsible and inconsiderate. An example is the brown bag seminar I was scheduled to give on 17 February. In January, he first cancelled the seminar, but then he agreed that I was to give it. Two hours before the talk he called me to say that he was canceling it again. His reason was: I cancelled it because of the reorganization in PCs.’ However, Jobs did not tell the seminar’s organizer about the cancellation, nor did he place any notices announcing the cancellation.”

“At noon, fortunately, I made a last-minute decision to go over to the seminar site, where I discovered a crowd of over 100 employees waiting to hear the seminar. I announced the cancellation myself - and then I gave a talk on my current work and interests at Apple, instead.”

I was careful not to mention Macintosh or give specifics since Jobs had forbidden it, but just explained the cognitive aspects of the interface and design principles my group and I had developed; it was - as everybody knew - a veiled introduction to the Macintosh project.

The talk was received very enthusiastically. The morning after the seminar Jobs called me into his office and told me that I had violated his explicit instructions and was fired. Ignoring what he said, since he often spoke without thinking things through, I told him that I’d come back in the afternoon, after I’d completed something I was working on, and we’d discuss the matter.

Later I went back and, as I expected, he had decided not to fire me but I was “given” an extended paid leave from Apple. The leave turned out to be a very important time in my life. For one thing, I went to a party at marketer Joanna Hoffman’s house where I met my future wife, Linda Blum. I found a place to live on ten acres of land high in the foothills of the Santa Cruz mountains, offering a magnificent view of Silicon Valley. I rebuilt the dilapidated old house that stood there, adding a large music room for my piano and a small flying field for my model planes.

When I came back from my leave I was offered the position of head of Apple’s research division. I had been offered this before, had accepted, hired a good group, and seen them whisked away to “put out fires.” In those days Apple didn’t know what research meant, and looked at the talented people I hired as resources wasted if they weren’t working on current products. Besides, there was the matter of personal integrity. Steve Jobs had become impossible for me to work with.

Most people worked around him or sucked up to him or were in awe of him. In fact he was no genius; he resembled a planet shining by reflecting the light of others. Yet he thought of himself as the Sun King. He could not abide someone who was unimpressed by Steve Jobs, yet by his actions he had lost my respect, and I am incapable of being a sycophant.

Steve had *chutzpah* in the extreme; he said that the Mac would make “a dent in the universe,” without the least idea how big the universe is, or how little a dent all our activities really make. And you can also explain Jobs with another Yiddish word, *mensch*. It is high praise to say of a person that he (or, in these enlightened days, she) is a mensch or “a real mensch.” A mensch is cultivated without losing the common touch, upholds high principles while remaining practical, is kind and generous without short-changing himself, and is attentive to his responsibilities to himself, his family, his business, his associates, his community, and the world. If you understand the qualities that make a man a mensch, then you understand a lot about Steve Jobs. Everything a mensch is, he isn’t.

At this point the only alternatives left to me were to leave or learn to toady to Jobs. I resigned from Apple and gradually watched my predictions about the Mac come true. Jobs took until 1984 to get the project out. Burrell Smith quipped that it was in “constant time to completion mode” and I was repeatedly told that Jobs did exactly what he said I would do, make endless mindless changes; I have many faults, but lack of direction is not one of them.

Steve was given to imposing absurd requirements on the project’s designers. This was especially ironic since one of the arguments he used, to convince management that he should be given the Mac project, was that I was an “academic dreamer and a perfectionist” who would keep on changing his

mind and never bring the project to fruition on time. My detailed schedule showed release of the product toward the end of 1982. Jobs's verbal plan was something like six to eight months shorter.

There is some evidence to back up my perspective: the next project of comparable scope that I managed (a workstation for Canon) was completed on budget and on schedule. The next project Jobs tried to manage (the NeXT computer) was a disaster in both these regards.

The resultant Mac not only took more time to come to market, but was a less coherent and — in some ways — less capable product than what I had been working towards. The interface was less consistent and harder to use, and there was no way to get at the hardware bus. Other parts of the Mac design were improved. Whether my version (as it would have matured as it approached production) would have been more commercially successful than the 128K Mac is an unanswerable question. The experiment cannot be done, and we will never know. I feel it would have been a somewhat better product that would have penetrated the market faster. I would guess that Jobs would disagree.

DREAM FULFILLED... ALMOST

Three decades ago I dreamed of a computer with which I could compose music and print it out in full musical notation, write properly formatted text in a panoply of fonts, have the ability to mix text and graphics, and do drawings with precision and ease.

Today I do all this and more at the tiny and capable Macintosh PowerBook that sits at my desk. It goes wherever I do. This much of the dream came true.

My reasons for deciding to abandon teaching for commerce proved correct. The Macintosh's profitability (as contrasted to Xerox's extensive published record) convinced companies such as Microsoft and IBM that the interface was the controlling element in most sales. Now a vast majority of the computers sold have an interface that looks much like the Mac's.

Because of the Macintosh project, computing has been made easier and more pleasant for hundreds of millions of people years before it might have happened otherwise. I made not a penny for my work on the Mac, beyond my salary at the time;

but I helped change the world in accord with my own personal vision, and I have seen the effect of this in my own lifetime. This would be fully satisfying if I didn't know so well that we can do much better.

The desktop metaphor, used by everybody from PARC through Apple and Microsoft and extended almost absurdly by Apple spin-off General Magic, was a clever way of making the workings of an operating system palatable and learnable. It is far more fundamentally good to eliminate the need for an operating system altogether. The current paradigm of using application programs is inherently wrong from the standpoint of interface design. This is widely recognized, but the solution offered is to make them interoperable, which solves some of the problems but by no means all.

GUIs as presently designed and used are an interface dead end. They can be patched endlessly, but only a completely different approach can bring a large jump in usability. The Cat computer, which I developed for Canon, demonstrated that my alternate approach is implementable and both more productive and more pleasant than GUIs. Canon failed to market the product effectively, possibly because the moribund Electronic Typewriter Division had been selected for the task, and it is now a dead Cat.

The parts of computer interface design that I am working on now are not dependent on particular technologies; any advance in the basics of interface design will apply — however more powerful computers become, however broad the information networks of the future spread, and however the technology is melded into our everyday or even everymoment lives.

A CRITIQUE OF SOME HISTORIES

Many friends have suggested that I counter the numerous incorrect accounts of the history of the Macintosh with a true one of my own. It is difficult, for even though nobody was or could have been more closely involved with the initial creation of the Macintosh than I, I cannot eliminate the colors that tinge my memories. It was a very emotional time, full of strong feelings, massive egos in conflict, distinctive personalities, and many rights and wrongs. But I cannot do worse than some of what has been published.

There is a strange avoidance of scholarly seeking after truth. An egregious example of the anti-academic attitude occurs in a book by Robert Cringely, who writes a delightful column that appears weekly in *InfoWorld*. In his book he has the Mac and Lisa projects being created by Steve Jobs after Jobs made a visit to PARC in 1980 and came back inspired.

I wrote to Cringely and pointed out that his account - like those of several other authors - was wrong; Jobs had indeed made the visit in 1980 (some say in December of 1979,) but the Mac project was proposed in the spring and officially started in September of 1979. In other words, the project was well underway before the supposedly pivotal event took place.

Cringely was unabashed. He wrote back: "As for all the business of what project started when, whether Lisa started before or after Steve visited PARC, whether the Mac had already begun or not, well I don't think that it really matters very much. My attempt was to EXPLAIN (I say that at the front of the book), not to be a historian." How one can hope to explain what happened, without even knowing what happened, eludes me.

A PBS special on the history of computers made the same mistake of attributing the genesis of the Mac to Jobs's visit to PARC. When I wrote to Jon Palfreman, its producer at WGBH, he replied, "The part of the program you are referring to comes at the end of a lengthy segment about the highly innovative work done at Xerox PARC. This section was based on extensive interviews with Alan Kay, Bob Taylor and Larry Tesler. The purpose was to show that the key concepts of interface design which today are a feature of most PC's (if you count Windows) were first discussed at Xerox PARC. When those ideas were embodied in an affordable machine — the Macintosh — they began to change the world of personal computing. I was aware of your key role in the Macintosh project, and indeed of the contribution of people who developed Lisa. My aim in this particular program wasn't to detail the history of Apple, but to show how the key interface ideas found their way into consumer PCs."

Again the false scenario seems so plausible and story-like that the person in charge does not care to "detail the history." But it is in that history, and not only the history of PARC, that "the interface

ideas found their way into consumer PCs." The people he interviewed were at PARC; their association with Apple began only after the Mac was well under way. Thus they could only tell him about the development of the ideas at PARC and, in the case of Larry Tesler, about the work on Lisa only after 1980 — that is after Apple was committed to the basic direction I wanted the company to take.

Larry was quite resistant to some of the non-PARC ideas that we had developed independently. He did not at first understand many of the improvements over Xerox's work — such as the one-button mouse — that I created. He did not work on the early Macintosh project at all but on Lisa, which was modeled closely on the Xerox Star, even to the point of having the same Xerox-created names for the fonts. The Macintosh proceeded for years much more independently and (significantly for the reports that used them as sources) out of the view of the people interviewed!

The years of thinking and experimentation on the early Macintosh project have gone unreported, even though the early work led to the breakthroughs that made the Macintosh and everything after so much of an improvement over what went before. Against this reality we have the powerful mythological image of Jobs going to PARC, having an "Aha!" experience and coming back at full cry to Apple to create a fantastic project.

The fabricated Jobs story is familiar — it parallels that of Archimedes jumping naked out of his bath crying "Eureka!," and a dozen other stories. That there was a little-known computer scientist who had been working on the concept for over a decade — who created the project, and then maneuvered Jobs to go to PARC, so that Jobs would begin to understand (and thus support) what was already going on at Apple — is a very different, more complex, and unlikely-sounding story.

Also, the appealing and basically true legend of two college drop-outs who created the profitable and excellent Apple II blends in easily with the fiction that one of the dropouts went on to create the even more revolutionary Macintosh. It is a less striking tale that a former college professor and computer center director with a degree in computer science instigate such a thing — but although it may not be as good a story, it is what happened.

Cringely and Palfreman were not being underhanded, only a bit careless and — in Cringely's case - cavalier. In some other cases, authors drew the wrong conclusion by lacking accurate information. Jeffrey Young, in his book "Steve Jobs," writes of the first time that Jobs (along with Atkinson and others) saw the work at PARC.

"Atkinson and the others were asking Tesler questions, one after the other." "What impressed me was that their questions were better than any I had heard in the seven years I had been at Xerox... Their questions showed that they understood the implications and the subtleties..." But Young did not ask why their level of instantaneous understanding was so impressive. The reason was that I had been explaining all this stuff to Atkinson and Jobs for years; Atkinson (a student of mine who had worked with me extensively prior to this meeting) had grasped it very well. Tesler didn't know about this background, wasn't told, and so was bowled over.

Atkinson couldn't very well say that Raskin had briefed him and some others, because I was out of favor with Jobs at the time, and anything I proposed was automatically rejected. Only after much planning and sleight of hand, during which it appeared that Atkinson supported the PARC trip and I opposed it, did Jobs agree to go.

There is also the halo effect. During the years that Steve Jobs was on top at Apple - and before NeXT showed his fundamental weakness - he was usually credited with inventing the Macintosh. Later, when his star was declining — as his company, NeXT, beat one strategic retreat after another, and as General Magic, co-founded by Bill Atkinson, Andy Hertzfeld, and Marc Porat, was about to announce its first product — the December 27, 1993 issue of *InfoWorld* included a story hailing Bill Atkinson and Andy Hertzfeld as the creators of the original Macintosh.

Their contributions were essential to the product and represent some brilliant work, but neither of them has ever claimed that they created the Macintosh. Again we find the heroes of today falsely credited with the achievements of others not currently in the limelight.

Steven Levy's history of the Macintosh, *Insanely Great* — published to ride the wave of publicity for the 10th anniversary of the Mac — is also occasion-

ally at odds with historical fact. Levy retells the Jobs-at-PARC story. Strangely, he credits me with having paintings shown at a famous museum; in fact I have never done any paintings. (An adaptation of this book, published in the February 1994 issue of *Popular Science*, tells a story that is far more accurate, — although it still calls me a painter.)

John Sculley, in his ghost-written book *Odyssey*, refers to me as a "programmer" at Apple. I was never a programmer at Apple, and the rest of what he says is nearly as inaccurate. He got his misinformation about the history of the Mac primarily from Jobs, with whom he spent a lot of time. Like Cringely, Levy, and Palfreman, he chose never to interview me or even call me — or others who were there — to check on his facts.

My experience with Jeffrey Young was especially disturbing. I had agreed to the interview with the understanding that I would see and comment on the galleys before publication; he never sent them - his inaccuracies compounded by a breach of trust. By way of contrast, Owen Linzmayer's *The Mac Bathroom Reader* is more accurate; for example, it gets the order of events straight.

In any case, Jobs's own view of how things came about necessarily must have been distorted. For one thing, we often misattributed ideas deliberately when speaking to Jobs. If the group admired an idea by someone Jobs didn't like at the moment, we gave the credit to someone currently on Jobs's "good" list. It was also often necessary to use "reverse psychology" on Jobs; we got a lot of features into the Mac by having someone (usually me) suggest the opposite. Jobs would then see the problem in "my" approach and often tell us to turn it around.

Another technique was to tell him about something informally. Often he replied that the idea was dreadful. Then when he "proposed" that same idea after its merits had settled in, a few days or weeks later, we'd tell him he was a genius for having thought of it.

Thus, Jobs's recollections of the history of the Mac would often be far from what actually went on. Being very independent, I was often on Jobs's "bad person" list, so I had to rely heavily on these techniques. You'd think he would have caught on when one of "his" ideas turned up implemented later that afternoon or the next day, but he simply

believed that his great engineers — and his way of driving them — could get it done so quickly. He had little intellectual basis on which to judge the difficulty of software or hardware tasks, which often helped us pull the wool over his eyes. How it all looked to him I cannot say. Eventually his impossible management style became so well known that Sculley and the board of directors of Apple had to remove him from all functional duties in the company. My memo had finally been acted upon.

JOBS AND THE PIPE ORGAN

Here's another story that doesn't quite fit in anywhere, but gives some insights into the interpersonal dynamics of the time. When I first started working at Apple, Jobs and I would take long walks (probably like the much-reported walks he would later take with John Sculley). I remember giving him mini-lectures on the philosophy of science or the performance practices of early music. On one of these walks I shared with him my life-long ambition to own a pipe organ. I explained that the valves to the pipes in many organs were driven electrically, and I hoped to hook up an Apple II to one which would turn it into a modern player organ. (I published an article with details in *Byte*.) He asked me why I didn't have one; I told him that it was mainly a matter of space, and that there was a secondary consideration of cost.

Jobs had a suggestion: if I could find an organ I could afford, I should buy it and Apple would let me put it up in the lobby of the new, large building on Bandle Drive, in Cupertino CA. I would hook it up to an Apple II, which would play it for visitors. After hours, I could practice on it, or even give company concerts.

Jobs was excited about the idea and told many people about the organ that was going to be installed. With this encouragement, I searched for an organ in earnest. In a few months I got lucky and found an abbey where the organ was being replaced. I told Jobs the good news and he congratulated me on the find. I purchased their old organ, had it crated and moved onto the abbey lawn (no small task in itself) and called Jobs to tell him that the crates with the organ would be there in a day or two.

"What organ?" he asked. "The pipe organ we're putting up in the lobby," I replied, thinking that he must be distracted to have forgotten. He first said that he had changed his mind, because it looked like space would soon be tight; when I suggested that it was a bit late to change his mind, since I had already purchased the organ, he retorted that he had never agreed to have the organ installed at Apple in the first place.

When I got back I reminded him that he had made a commitment, and that I had gone to some trouble and expense based on his assurances. He told me that he had never assured me that he would give me room for the organ, and refused to speak in my presence to the people who had heard his promises. I asked if, until the issue was resolved, I could store the crates in the still-empty buildings. The crates were outdoors, this was an imposition on the abbey, and if it rained, the organ could be seriously damaged. He simply said no.

I was stuck. The organ was too large to fit into my house or even a rental storage unit. I called every organ builder, organ teacher, and church I could find and, after much desperate work, found a church in Santa Clara that needed an organ. They, in turn, found a benefactor to purchase the organ from me for them. After long negotiations I succeeded in selling it for a fraction of its value.

This was my introduction to the new Steve Jobs, or perhaps a phase of the old Jobs I hadn't yet seen. Apple's first employee and his friend of many years, Dan Kottke, who had traveled with him in India and worked with him day and night to help Apple get started, was treated even more shabbily. In late 1980 Dan was surprised to find out that — in spite of their long friendship and the many uncompensated hours he had put in — he was not going to get any stock options. Later Woz gave some stock to Kottke, and to some other deserving people from the early days of Apple (such as Bill Fernandez, Chris Espinosa, Randy Wigginton, Cliff Huston and Dick Huston) all of whom Jobs had turned his back on. Woz's was an admirable act of pure generosity.

As for me, I still don't have a pipe organ.

This is a preliminary version of a portion of a book in progress. Comments and corrections are welcomed. Please send them to jefraskin@aol.com.

SPEAKING OF ENGINES....

Joel Shurkin has updated his eminent survey text of computer history, *Engines of the Mind*, and W. W. Norton has done us all the favor of publishing the new edition in paperback. Whether you've read this book before or not, you may properly rejoice that it's available once again; one of the few respectable attempts to pack its topic into a single volume, *Engines of the Mind* is compelling, energetic and well-annotated. The new paperback is priced at \$US13.00 and its ISBN is 0-393-31471-5. Go hound your favorite bookstore.

FTP SITE FOR COMPUTER CONSERVATION SOCIETY

by Chris P. Burton

The Computer Conservation Society, a special interest group of the British Computer Society in association with the Science Museum of London and the Museum of Science and Industry of Manchester, now has an ftp archive site.

All issues of the Society quarterly bulletin, "Resurrection", are available in several formats, as well as a small collection of simulators of historic machines. Many of these simulators have not been available previously, and more are "in the pipeline", awaiting documentation from their authors.

To access the archive, make an ftp connection to <ftp://ftp.cs.man.ac.uk/pub/CCS-Archive>.

World Wide Web pages will be announced at a later date. Meanwhile, readers may be interested in WWW pages of the work on Colossus at Bletchley Park, where the CCS has an exhibition room. The URL is <http://www.cranfield.ac.uk/CCC/BPark>.

THE FRIDEN EC-130:

The World's Second Electronic Desktop Calculator

(With some notes about the world's first.)

by Nicholas Bodley

BACKGROUND

In late 1963, advances in electronic technology made it practical to build and market an electronic desktop calculator. The world's first was the Anita, made by Sumlock Comptometer (Ltd.?) in England. The Anita had a beautifully built full keyboard; each digit place had a vertical row of 10 (or 11) keys, in a row extending toward and away from the operator. It probably had 10-digit input, which would imply 100 keys for digit entry. (The eleventh was a column-clear key, most likely; a zero didn't need to be explicitly entered.)

The display used shaped-cathode neon glow numerical-indicator tubes, known among the technical community in the USA as "Nixie" tubes; this is a trademark, probably of Burroughs. The internal logic used beam-switching decade-counter tubes; one variety (not necessarily that used in the Anita) had thirty cathodes. Only one cathode would have a glow discharge; a series resistor in the anode circuit kept the voltage below that required for other cathodes to conduct. Ten cathodes kept the count; the other twenty were connected into two groups of ten, and a two-phase clock stepped the glow to the next "stable" cathode in two stages by temporarily "stealing" the glow from the "stable" cathodes.

The Anita performed all four functions, but because each digit was costly, it wasn't designed to display all digits of a full product of a multiplication. This wasn't too bad if the operands were integers; but it did have decimal-point logic, and the decimal point sometimes appeared in a bizarre place when mixed operands were multiplied — it wrapped around the end and popped up in a logical, but peculiar, place. Speed was of no particular consequence; an "all nines" multiplier imposed no special wait.

About a year after Friden introduced the EC-130, Marchant brought out a nice, rather compact machine called the Cogito, with a display like that

of the Friden but with peculiar half-size zeroes that looked quite odd.

Monroe was the other of the "Mechanical Big Three" in nonprinting desktop calculators in the USA, but seemed late in bringing out an electronic desktop machine; however, they had a brilliantly designed mechanical printing calculator, in two units cabled together, that was a successful competitor to the EC-130. The model name was PC-1421.

THE FRIDEN EC-130

Having sketched in the contemporary competition to Friden, we can proceed to recollections concerning the design and success of the Friden EC-130, a very significant product in the history of calculators. The author, Nicholas Bodley (nbodley@tiac.net as of this writing) was one of the original eight technicians to be trained at the factory on this calculator. It was fascinating, exciting, and completely memorable to see this device for the first time; the EC-130 was beautifully styled although, in my opinion, some of the prototype's striking quality was lost in the translation to production dies.

EXTERIOR PACKAGING

In external appearance, the EC-130 was a distinctive, rather low box with a gracefully curved top cover over the electronics, and a vertical panel (mostly blank) with an extended glare shield around the display face. The keyboard extended across the full width of the machine; it was a 10-key, serial-entry type, with various function/control keys on both sides. The keyboard panel sloped up toward the vertical front panel. The rear was a finned die-cast heat sink of generous proportions, although power consumption was modest. Overall "footprint" was that of a fairly large electric typewriter, although perhaps longer from front to back. Friden's chief engineer, the wonderfully capable Robert Ragen — one of the most brilliant people I have ever met, and a pleasant, rather self-effacing fellow in the bargain — was responsible for the remarkably innovative architecture of the EC-130, which was produced at a 1963 list price of roughly \$2,100.

ARCHITECTURE

When the EC-130 was designed, ICs were hopelessly expensive; I recall that a Westinghouse DTL-930 NAND gate, with perhaps four inputs, cost a big chunk out of \$US50 each. The EC-130 design, in a militantly cost-conscious setting, was all discrete PNP germanium diodes and transistors. (To duplicate the internal logic of one of today's calculator chips with discrete components would — at a personal, rough guess — result in a box too big for a desktop, which might draw a few hundred watts, and cost about \$10,000.)

Reverse Polish Notation (RPN)

This calculator preceded Hewlett-Packard's desktop machines, and was the first to use RPN, which Friden called simply "PN". (No, HP wasn't the first to use RPN, and they have never claimed so.) The EC-130 had a four-register visible stack functionally very close to that of the HP-48, for example. A stack drop caused zeroes to enter the top. It even had a "Last x" register, for repeat multiply, but the contents were not displayed. The Enter key worked exactly like that on the HP calculators.

DISPLAY

Output was displayed on a type 5DEP1, 5-inch round green-phosphor electrostatic-deflection CRT, with about 2 kV accelerating voltage. (All computer CRTs use magnetic deflection.) The characters were seven-segment, similar in appearance to those now universal in inexpensive LCD calculators. The beam of the CRT was swept across the screen to write the strokes that made up the individual characters; it was a vector scan, not a raster scan. The deflection waveforms were wondrously complicated, and always the same. (The slant of the characters was created by just one resistor that cross-coupled the vertical deflection into the horizontal, without a buffer amplifier; it was a marvelously simple yet subtle circuit. There was no visible cross-coupling the other way, and it didn't seem obvious from looking at the circuit how the isolation was done.) Individual digits were created by unblanking (turning on) the CRT beam at the appropriate times; the decoding matrix used a remarkably small number of diodes, probably

about 80. There was a decimal point visible in each register.

DECIMAL POINTS

Marketing decreed that logic for a floating decimal point, universal in contemporary calculators, would be too elaborate for inclusion in a discrete-component design. The EC-130 did have decimal points and a decimal-point-entry key, but the display had a selectable fixed point, the same for all registers of the stack. The user had to decide how many decimal places to work with, then make the selection through a rotary switch with an edge wheel knob projecting through a slot; about six choices were available. I have seen two different sets of [numbers of places] in different machines. Internal logic permitted any number of places within the limits of the machine, but the switch was the constraint. Someone must have hacked a freely-selectable decimal selection at some time.

The machine had 13 digits and, through repositioning of the decimal point, could provide all 26 digits of a product. Overflow and/or truncation naturally occurred in such cases. It could work as a purely fractional or purely integral machine. Division required 1.050 seconds for an all-nines quotient; an all-nines multiplier was slightly faster.

KEYBOARD

The keyboard was serial-entry of the type called 10-key. Its unique mechanism was borderline practical as a design. Each keystem was part of a stamped piece of steel, which included a "blade" with a rounded bottom edge extending from front to back. Pressing a key moved this blade down against ramp-shaped recesses in seven (or eight?) code bars made of stamped phenolic laminate and positioned crosswise.

Each code bar had a small magnet attached to it, which operated a reed switch. As I recall, the code bars had no return springs but were actuated positively in both directions. One code bar was actuated for any of the ten digits, and reset for other keys, resulting in the terms "common function" and "common digit" frequently used by the tech. folk. There probably was one code bar reserved to distinguish digits from non-digits. A small, fast electromagnet with an armature locked the code-bars in place — and the keys as well, in some way,

— until a time-consuming operation such as multiplication or division was complete. It always was pulsed, but most operations were quite fast, and it served as a keyclick noisemaker most of the time. Some malfunctions could lock the keyboard; it wasn't rare for a defective machine to have its multiplication key, once pushed, lock down and stay.

All functions and digits were coded, with codes that specifically allowed minimal component count in the electronics. The "touch" was quite acceptable despite the oddity of the design.

INTERNAL STORAGE

Like all desktop calculators of its time, the EC-130 required too many digits of internal storage to allow use of discrete-component flip-flops for data. Static RAM ICs came quite a few years later, even after the heyday of serial shift-register ICs. (One Toshiba machine that came out a few years later, the so-called TOSBAC, stored in discrete capacitors with refresh circuits; the basic principle was that of dynamic RAM! Shades of the Atanasoff-Berry machine...)

The Friden machine stored data in a low-cost implementation of the ultrasonic wire delay line, a truly serial storage. These devices had been used in expensive systems, with tight control over delay time and clock frequency, for mass storage of binary data; apparently the idea was to make the delay time some large multiple of the clock period, and control it to within a fraction of one period. Friden took a simpler approach. Once all the digits had been clocked into the delay line, the timing chain (cascaded binary counter stages) did a carry (overflow) out the most-significant flip-flop, and shut down the clock — so to speak, although the oscillator continued to run.

The first pulse written onto the line was a dummy pulse; when it came out of the line again, after a delay of four milliseconds, it effectively restarted the clock. Short-term drift was accounted for, and kept within good bounds. The timing chain had three fast stages run by the oscillator; these were, as I recall, jammed to zero by the start pulse, to force the apparent clock phase to be in step with the start pulse and succeeding data. (This technique is probably embedded inside every UART chip.) The actual clock rate — not the oscillator rate — was 330 kHz.

Since pulses on the line were retimed and rewritten with every "latest" clock frequency, modest medium-to-long term drift was of no consequence. A fairly large timing gap, between the last digit written and the start pulse, allowed for both long-term droop in clock frequency and mechanical tolerances of the delay line. (The oscillator used a discrete molded inductor and maybe a couple of silvered-mica capacitors, perhaps a Colpitts circuit. Quartz crystals weren't needed, and ceramic resonators were probably 25 years in the future.)

The line itself was a subassembly on the bottom of the calculator, a flat spiral of about eight inches (20cm) radius and roughly a dozen turns. The wire itself was mild steel; it was carefully selected, but no exotic alloy was needed. The spiral, supported by soft silicone rubber sheets with punched holes and loading slits, sat in a shallow sheet aluminum tray/chassis with a huge hole on the center.

This type of delay line stores torsional pulses with a duration of a very few microseconds at most, and an angular magnitude (probably) substantially less than one degree of arc. Even given the short duration and small magnitude, stress on the wire was probably relatively high. Several thousand such pulses could be launched into this wire, and remain adequately discrete at the far end. (Whether soliton phenomena are involved, I don't know, and probably the original designers didn't, either; I suspect solitons are a comparatively recent discovery.)

Pulses were launched with magnetostrictive tapes, probably of pure nickel, and welded carefully to the exact end of the wire —tangential to the surface, and at right angles to the length, of the wire. (If you hold a pencil between your thumb and forefinger, and let it droop, the tapes are your fingers and the pencil is the wire.) My recollection is that each side of the wire used two tapes for engineering reasons.

These tapes were passed through the bobbins of two tiny coils, positioned close to a permanent bias magnet. The magnet's field made the tapes shorter, by perhaps a few parts per million, than they were without the field. Pulses to the coils canceled the field for one tape, and doubled it for the other. In probably several hundred nanoseconds, the longitudinal stress pulses traveled to the end of the delay wire and gave it a sudden twist, followed a microsecond or two later by a relaxation to normal.

Stress pulses in the tapes would also travel the other way, but these superfluous pulses would reflect from the ends of the tapes, and be absorbed by sheets of silicone rubber which also functioned as a support.

Later printing calculator designs (with no display, just a vertical stack of four lamps to show non-zero contents) used IC logic and a smaller-diameter coil; tapes were "single-ended" rather than of the earlier push-pull type. Apparently they worked quite well enough. These machines had the Singer logo on them.

At the other end of the delay wire, a second similar transducer converted the torque pulses to longitudinal ones; the inverse magnetostrictive effect, fed into another permanent magnet, developed adequate signal in a second pair of coils.

Pulses coming off the line had a shape reminiscent of the wavelet sombrero function. (I recognize that there may be a more formal name for these.) The pulse takes a negative-going rounded dip, returns and crosses the zero axis, continues to a peak maybe three or four times as high as the dip, and then falls back to a second dip like the original. It settles quickly afterward. Simple gain stages and a slicer convert such a pulse to a clean rectangle at logic level.

INTERNAL DATA REPRESENTATION

The internal representation of the digits was *not* BCD; it wasn't really coded at all, but became radix-one on the wire. Digits one through nine were represented by strings of pulse count equal to the digit being represented. (A two was two pulses; a nine, nine pulses.) Each digit was given its own time slot in the total data time of 48 μ sec. A lack of pulses occurring at a given time was interpreted as a zero digit.

Remember that the timing [counter] chain was started by the first pulse out of the line after an "end-around-carry" stopped the counters. This, combined with good short-term stability of the clock, meant that the "number" in the counter chain defined the identity of the digit.

I no longer recall how many internal 13-digit numbers were kept in the calculator. Four stack registers were all displayed; two others, not displayed, were a store/recall and an arithmetic register. One was a "last x" register with space to store

one 13-digit number, and there were Store and Recall keys.

Numbers were written onto the delay line in a sequence with the "hidden" registers first, then progressively up the stack; all LSDs were written before the next digit. Decimal points were not stored since their location made no difference in addition and subtraction; multiplication and division decimal settings simply affected where the result digits were placed.

Once in the electronics, the digits took on another unusual form, but before I explain that I'll pop the real surprise: *There was no adder in this calculator*, or at least not in the conventional sense. You might well expect some such coding as 8,4,2,1 BCD, or excess-three, or 4,2,2',1, with a 4-bit combinatorial adder. But no....

5 = 11111

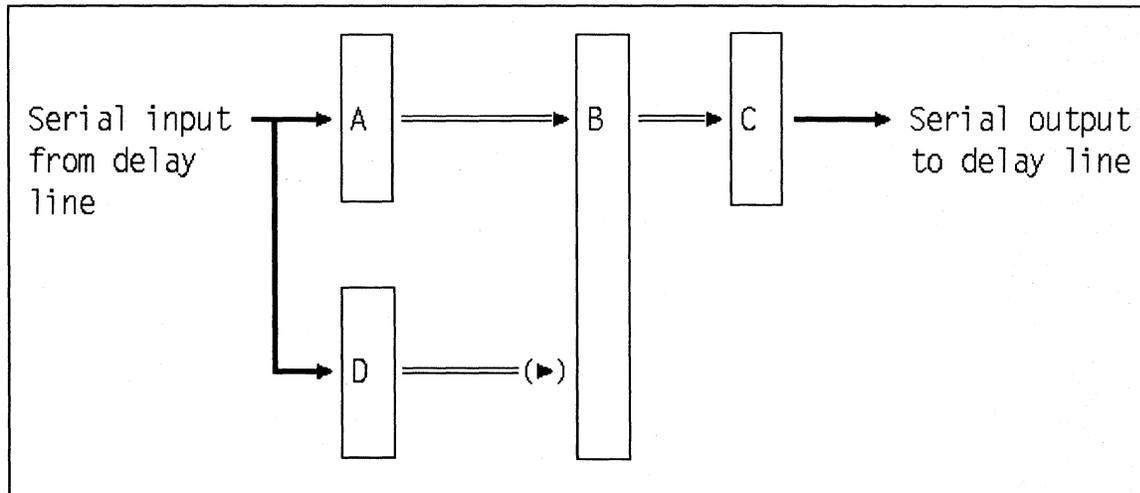
6 = 11110

7 = 11100

8 = 11000

9 = 10000

This code is "wasteful" in that 22 of 32 states are disallowed, but in the calculator, it permitted a fast bit rate on the delay line and use of particularly simple, low-cost logic to count the pulses coming off the line. This logic is called a Johnson counter or "switch-tail ring counter;" it's basically a 5-bit shift register with a parallel reset for all stages to set it to zero. The normal and complement serial outputs are connected back to the inputs, but with a half-twist, so that a zero going out one end shifts in as a one at the other.



Flip-flop arrays

(I'm not sure, but I think [A..D] are Friden's designations.)

In the electronics, digits were represented by a code related to the Morse code for each digit. It is and has been used elsewhere and there are (or were) ICs that counted in this code. It's a really simple nonweighted code, which we might call the *baklava code*:

0 = 00000

1 = 00001

2 = 00011

3 = 00111

4 = 01111

The code bars in the keyboard defined the ten digits by this code; "common digit," when asserted, signified that the code was to be interpreted as a digit. At the proper time, this digit was read into the arithmetic unit in parallel.

ARITHMETIC UNIT

The arithmetic unit contained three counters of this type, and a simple 5-bit register with parallel input and output. This totaled to 20 flip-flops, not an insignificant circuit when built of discrete components. This shows the arrangement of the four 5-bit flip-flop arrays, with one block for each group of five.

In this illustration, the single lines are serial data paths; the inputs to the A and D counters are logic pulses from the delay line's read amplifier. These

pulses (when enabled by gates, which is usually) cause the A counter to count up, and the D counter to count down. (Subtraction!) Likewise, the C counter feeds pulses serially to the delay line's write amplifier until it counts down to zero.

Double lines represent 5-bit parallel data paths; when these are enabled, the contents of a given counter/register shift to the right, into the "new" counter/register. This happens every time a digit comes out of the delay line.

When the calculator is just sitting there, for each digit, the A counter gets reset; then the serial pulses from the delay line's read amplifier make it count up. (The D counter probably counts (down) as well, but nothing is done with its contents.) At the end of this time slot, the contents of A shift in parallel into B. The next digit time, the digit in B shifts into C. The next digit time after that, clock pulses to the C counter make it count down and feed serial pulses to the delay line until it counts to zero.

The D counter's output is usually gated off; it doesn't go to the B register. (That's why I put the > in parentheses.)

To do an addition, the A counter is not reset when it usually would be, and the next digit from the delay line causes its count to increment "on top of" the count already there. If it counts past nine, logic detects the fact, and a carry flip-flop is set. (Extra logic accounts for carries caused by a carry.) At the next digit-column time, the logic increments the A counter by one, to add in the carry. Similar things take place during a borrow in subtraction.

There is at least one path not shown for parallel transfers. My memory of this isn't clear, but I think a bypass around the B register feeds the contents of the A counter directly to the C counter. This makes a digit return to the line, but one (register) time earlier than usual. The end result is that the stack drops. Stack lift involves an extra delay; I think the D counter serves as a plain register (i.e., not a counter) for that, again using paths not shown. A given digit would probably take the path (line) to A to D to B to C to (line).

Subtraction, as I said "prematurely", involves decrementing the D counter; it's connected/ defined as such. I really don't recall the details, but they're not any great mystery. None of the three counters is bidirectional.

Multiplication involved an extra (time-defined) 13-digit register, and might even have involved shifting out multiplier digits from one end, then product digits into the other; the MQ registers of 1970s-era computers come to mind. The D counter might well have controlled the number of addition cycles. The control logic, while fairly complicated, was no more so than necessary — every resistor, capacitor, diode, and transistor counted when determining cost.

Division, as expected, performed consecutive subtraction (and tallied the number of subtraction cycles) until underflow, following with a restoring add-and-shift.

CONTROL

Control was of the state-variable variety. I dimly recall a 3-bit control counter that advanced when the next detailed stage of an operation was ready to take place. There was no explicit diode matrix in one spot for decoding logic states along with the contents of the counter; the logic was simply embedded. When square root was added (vff.), the control counter possibly had to gain another bit.

CIRCUITRY

Most transistors were 2N1305's, germanium PNP, in TO-5 cans, except where the faster 2N1499's (TO-18) were required. The deflection output stages were 2N2043A's, a popular RF power transistor. Logic diodes (DTL) were 1N662's. The circuitry was highly optimized; collector load resistors and base resistors, all 1/2 watt Allen-Bradley carbon comps, were individually calculated. Capacitors were chocolate-colored dipped mica, mostly, and also individually "calculated".

The CRT's high-voltage power supply was an inverter; it had a small transformer like a low-powered vacuum-tube filament transformer connected "backwards", and a voltage multiplier chain. The main logic power was quite conventional. There were roughly 165 logic gates, perhaps 40 flip-flops in all, of which the timing chain had 14. In total there were 300 transistors.

Of seven circuit boards, each about 12 inches by 5 inches (30x13 cm,) six were paired with spacers and (amazingly) lots of hand-soldered jumpers to join the two at the edges opposite the connector edge. One connector served both boards in a pair; the A

through D counter/registers occupied either one board or one pair. Layouts were rather closely packed. The boards mounted vertically and plugged downward into edge connectors, in standard fashion.

The EC-130 no doubt served as an inspiration for the Hewlett-Packard desktop calculators; I remember how amazed I was to read, in the *Hewlett-Packard Journal* of the time, about their scientific desktop machine, which had no ICs either! That HP machine, the 9100A, was quite as remarkable in its own right as was the Friden.¹

MANUFACTURING

Circuit boards were double-sided, with plated through-holes, and plating was done after drilling as usual. The pattern of traces was "deposited" by reverse electroplating in a unique machine developed by the inventor Gilbert Marosi. A copper negative master for the trace was wrapped around a drum and clamped. The drum was rotated in close registry with the drilled and plated copy, while electrolyte was flushed through a very small gap between the master and the copy. The liquid went in clear, and came out blue. The copper was immediately extracted and the electrolyte reused. The current must have been quite high, perhaps hundreds of amps. Any remaining copper that had disconnected itself electrically was removed with a quick etch. When this system worked, it worked quite nicely.

Unfortunately, there was one inherent (and nasty) contradiction. The process of plating the through-holes created faint anthill-like raised regions around the holes, which were of no consequence to the boards themselves, but they shorted out the Marosi machine. The result was significant damage to the master; apparently it wasn't easy to stop the current before the damage was done. (Multi-hundred-ampere transistors were still in the future.)

Manufacturing was, one would presume, in a major quandary. They had no practical alternative to the Marosi machine; setting up a conventional photoresist/etch line wasn't a job of a few days. They decided to "level" the unpatterned boards by following the plating step with a surface grind. The

shop knew more about surface grinding copper than I do, but not quite enough, and the consequences were horrible. The circuit boards would work fine until they were shipped to the branch offices, even sometimes until they reached the customer's premises. Then the failures would begin as the copper, fatally weakened by grinding, broke and separated the plating inside the hole from the trace on the surface.

At this time I was the "depot" service technician for the Northeastern United States. A torrent of failures descended on us, about 98% of which were mechanical intermittents that necessarily resulted from surface grinding; we had commendably few actual part failures. We would get a set of boards in, plop them into a test machine — stock, but with extenders for all the boards — and start warping the boards. As connections began separating we saw some wondrously peculiar symptoms, much more fascinating than those typical of a more conventional architecture.

If we found more than, perhaps, two definite symptoms, we opened up the pairs and I hand-soldered every through-hole on every board with great attention to technique. I still recall a gold-plated transistor lead that was surrounded by a perfectly good-looking fillet of solder. The lead went through the board to a pad on the opposite side, which in turn connected to some other circuits; but the lead was not connected to the pad on the component side, even though it was surrounded with a perfect-looking fillet. There must have been a very thin layer of rosin or other insulator on the lead, and the component-side pad had broken away completely from the plating in the hole. Solder continuity from bottom to top didn't help in this case.

THE FRIDEN EC-132

After I left depot maintenance on the EC-130, Friden introduced the EC-132, which offered a different selection of the number of decimal places, and added square root. Square root was considered a key feature by Friden; their mechanical Model SRW did square root using the "fives" algorithm, and their mechanical masterpiece, the Model SRQ, did both squaring and square root. Squaring was trivial on the EC-130; I suspect that the EC-132 also had a squaring key.

¹ See ANALYTICAL ENGINE 2.3, p. 6ff.

The internal architecture of the EC-132, as the model number implies, was an elaboration of the EC-130's, but the algorithm was probably new to that machine. More modern calculator architectures aside, at the time of the EC-13x it made sense to calculate square root by elaborating the division logic (or algorithm) to perform a division-related process whose "divisor" is constantly incrementing in a controlled fashion, instead of remaining static. The details of square root calculation by this method become somewhat messy, but the general scheme for incrementing the "divisor" is worth describing.

It's been known for a long time that the sums of the odd integers are the squares of integers, as a few examples will make clear:

$$1 = 1 \text{ squared}$$

$$1 + 3 = 2 \text{ squared}$$

$$1 + 3 + 5 = 3 \text{ squared, etc.}$$

This is the straightforward basis of "direct" square root calculation; the messy details involve shifts and preserving existing digits of the root as they develop, along with the developing "divisor", which gains another digit every time a root digit is found. (When I say "direct", I have in mind Newton's method, which progressively refines an initial estimate. That algorithm was thoroughly impractical for a calculator of the EC-13x era, and is probably not a good one for any calculator. I could be wrong, but what's taught in school sometimes differs wildly from the Real World.)

In the EC-130, incrementing a register by twos isn't easy at best; it becomes hairier still when the developing root digit becomes so large that a 9 must be incremented to an 11. The two-place representation of the current "divisor" digit becomes especially messy. Friden came up with an alternate scheme; people aren't generally in a blinding hurry for their square roots. Division involves repeated subtraction of a constant divisor; square root, as noted, increments the "divisor". Friden decided to subtract 1, first. If no overdraft, then subtract three; but, to create three, they didn't simply subtract three. They subtracted one, and then one more in a second cycle. This "one more than that" scheme (my term), involving a pair of subtractions for each increment of a root digit, was the key.

Each square-rooting cycle of "division" was now elaborated into a pair of subtraction cycles in each of which the "divisor" was incremented by one. Incrementing became much simpler as a result.

So:

Let's assume we want the square root of 16.

$$16 - 0 = 16 \text{ (overdraft not possible)}$$

Subtract one more than (zero): Take

$16 - 1 = 15$. This has now effectively subtracted 1. If no overdraft, increment the "divisor", which now becomes 1. Also tally one count for the root digit.

Now, do this:

$$15 - 1 = 14$$

and subtract "one more than" the 1,

$14 - 2 = 12$. This pair of subtractions has effectively subtracted 3. If no overdraft, again increment the "divisor", which becomes 2, and continue. Increment the tally for the developing root digit.

$$12 - 2 = 10$$

then subtract one more than 2:

$$10 - 3 = 7$$

This pair of subtractions has effectively subtracted 5. There's still no overdraft, so increment once again, to make the "divisor" become 3. Increment the root digit again.

$$7 - 3 = 4$$

and subtract one more than 3,

$4 - 4 = 0$ No overdraft yet, so increment the root digit once again.

There have been four pairs of subtractions; each time a pair created no overdraft, the tally was incremented, and now stands at four. The "divisor" also equals four in this instance; the calculation is complete.

Thus, incrementing by one with proper controls can calculate by the "consecutive odd integers" method. This case is simplified; in real life, that nice zero remainder would probably be ignored, handy as it looks, because it's a rare case. Most square root calculations leave a remainder (although no calculator I know of ever makes that remainder available to the user; it wouldn't be of

much use). In real life, the incrementing process would continue, the subtraction would cause an overdraft, and the overdraft would tell the control logic to get ready to calculate another root digit. The best information I have is that the EC-132 always computed to a zero result.

In the logic, that repeated "one more than" probably was represented by an extra pulse fed to either the A or the D counter after the digit in it was passed on to be rewritten to the delay line. I'm speculating here.

AFTERMATH

My boss, a decent fellow, finally had me transferred to Friden R&D in Rochester, which is another interesting tale I may tell someday. Somewhat sad; lost opportunity, but also mismanagement.

R&D was an instance of youth being wasted on someone young, and ended (back in 1966, I think) with early corporate downsizing/R&D cutbacks. It hit me early! Singer had bought Friden about the time I first went to work for them, and progressively destroyed a very good company.

One particularly interesting highlight of my time in Rochester came in 1964, when Friden was unofficially developing an electronic Flexowriter. It retained the basic electric typewriter mechanism, and even had a bus architecture, something quite advanced at that time. I wasn't directly involved with the project. It would have considerably extended the product lifetime of the Flexowriter, but management was, once again, fatally reluctant. The prototype added to the pain by being built with multipin connectors that looked great, but turned out to be miserably unreliable. The resident genius, Dave Frick, independently conceived of static RAM chips, and perhaps dynamic ones as well. This was when the first prototype dual-inline packages appeared — and it looked so stupid to standardize on a package that could be inserted backwards....

What Was The First PERSONAL COMPUTER?

An Exploration

by Roy A. Allan

DEFINING TERMS

Recent literature is inconsistent in defining "the first personal computer" as the term is presently understood. This has resulted, to an extent, from limited awareness of some early products with a claim to the title. Which computer was truly the first to be personal in the modern sense?

This article does not discuss the early personal use of mainframe computers; it also excludes experimental computers, such as the MIT Memory Test Computer, and minicomputers, because they were not oriented to a consumer market. In an article entitled "Is There Such a Thing as a Personal Computer?" [7], Lawrence I. Press presented several criteria — such as architecture, software, physical characteristics and marketing — which would serve to distinguish a personal computer.

Thus, for the purposes of this article, the term "personal computer" requires clarification. Up to the mid-1970's, when the majority of computers were mainframes shared by many users, a "personal" computer was defined as being designed for use by one person. As the technology progressed, our understanding of the term has changed; we now consider some early hardware too large and too expensive to qualify as being "personal," and require that "a PC" should also be small, easy to use, and of relatively low cost. Today's personal computer has evolved into a desktop appliance, available through the consumer market, that provides ready and affordable computing power to an individual. In this context, which of the early computers then called "personal" should we consider for the distinction of being "the First"?

1957: The IBM Auto-Point

John L. Lentz at IBM developed a small Personal Automatic Calculator (PAC) in the late 1940's, and described an engineering model of PAC in December 1954. This project evolved into the IBM 610 computer, described by the reference work "IBM's

Early Computers" [1] as being "IBM's first Personal Computer." The Model 610, also called the Auto-Point, was announced in September 1957, with a purchase price of \$55,000. The computer system consisted of a floor-standing cabinet — incorporating the electronics, magnetic drum, plug-board, and separate paper-tape readers and punches — with a keyboard for input and an electric typewriter for output. It was not a stored-program computer; a programmer entered instructions from the keyboard, or input them through the paper-tape readers.

Scientists and engineers used the Auto-Point computer to solve small scientific and engineering problems. IBM built about 180 units, but was not then an aggressive competitor in the development of small computer systems. Other organizations, such as Bendix Aviation, Librascope, Digital Equipment Corporation (DEC) and the Massachusetts Institute of Technology (MIT), were thus able to participate in an emerging market and developed, respectively, the G-15, LGP-30, PDP Series and LINC small computer systems.

1963: MIT LINC

Gordon Bell, a principal in the design of early DEC PDP-Series minicomputers, has stated that the MIT LINC was the first personal computer. MIT developed LINC — an acronym for "Laboratory INstrument Computer" — to facilitate the use of computer technology in biomedical research laboratories [3]. Principal designers were Wesley Clark and Charles Molnar. MIT demonstrated a prototype in early 1962 and completed sixteen units by mid-1963, which were assembled by scientific users to improve their understanding of the system. LINC had four console modules, an electronics cabinet and a keyboard. The processor logic circuits used transistorized circuit modules from DEC. An oscilloscope module could display a 512-by-512-point image. Memory was magnetic core with a basic capacity of 1,024 twelve-bit words; two magnetic-tape drives provided additional storage. Each system cost about \$32,000. Initial software was a text editor, an assembler and some utilities. A small number of scientific laboratories used the LINC computer in dedicated applications.

The IBM 610 and the MIT LINC were "personal" computers developed with reference to established

larger architectures, at a cost which confined their use to major scientific organizations. In the late 1960's a company called Computer Terminal Corporation (CTC) took the opposite tack by starting development of a "smart" computer terminal which would incorporate a microprocessor.

1971: Datapoint 2200

An *Invention & Technology* article [11] quotes Jack Frassanito as saying "I invented the personal computer..." CTC hired Frassanito in 1969 to develop a computer terminal that would have its own processor and other circuitry which would allow it to mimic other major computer manufacturers' terminals. The self-contained unit included a 12-line display, keyboard, two cassette tapes, a Texas Instruments bit-serial processor supported by TTL logic, and 8K bytes of internal memory. CTC introduced the resulting Datapoint 2200 in June 1970, and shipped the first units in early 1971.

Although CTC designed the unit purely as a terminal with unprecedented capabilities, a number of commercial users wrote programs and used it as an early, small, desktop computer. The Datapoint 2200, however, was still not made available at a price attractive to an individual user. The first microprocessor-based computer affordable for the general public was the comparatively unheralded Kenbak-1.

1971: Kenbak-1

An Early Model Personal Computer Contest, sponsored by the Computer Museum of Boston, ComputerLand and CW Communications in 1986, selected the Kenbak-1 as being the first personal computer. The criterion for selection of the winner was "interest, significance and date of each model" [5]. Designed by John V. Blankenbaker and introduced in late 1971, the computer used 130 small and medium-scale integrated circuits and had a memory of 256 8-bit bytes; it processed 1000 instructions per second. Input and output were very limited, with no keyboard or screen. Blankenbaker sold only 40 units at a price of \$750 [4].

The Kenbak-1, with its primitive user interface, stands in sharp contrast to history's next "first personal computer" — the stunningly innovative, feature-rich and costly Alto, developed by the Xerox Palo Alto Research Center (PARC) in California.

1973: Xerox Alto

According to Alexander and Smith's "Fumbling the Future: How Xerox Invented, Then Ignored, the First Personal Computer" [9], Xerox developed the Alto computer as a research project in the early 1970's; construction of the prototype was begun in November 1972 and completed in April 1973. Lead engineers included Chuck Thacker, Larry Tesler, Butler Lampson, Peter Deutsch, Bob Metcalfe and several others. Intended sale price of a system was about \$30,000, and the production cost of early examples was nearly that high. An Alto computer system consisted of a main tower case intended to fit under a desk, cabled to a grayscale bit-mapped display screen, 8 inches horizontal by 10 inches vertical, that could display 60 lines of 90 characters — a full portrait page. Input was by keyboard supplemented by the "mouse" originally designed by Douglas Engelbart at SRI. The processor was a 16-bit custom-made unit and basic memory was 64K 16-bit words, expandable to 256K. One or two 2.5-megabyte pack-type hard-disk drives were installed in the main cabinet. PARC also developed a new interactive programming language called Smalltalk which was used to create a windowed graphic environment and the desktop metaphor that was an unprecedented synthesis of hardware and software. Furthermore, PARC surrounded its individual Altos with a resource pool that included an Ethernet network, shared laser printers, and electronic mail. These were new human interface concepts that eventually formed the basis for developments by Apple Computer, Microsoft and others.

But the Alto, like computers previously described, and like other contemporary systems including the National Radio Institute NRI 832 kit, Hewlett-Packard HP 9830A programmable calculator, EPD System One computer kit and IBM 5100 portable computer, still relied on discrete components. Use of discrete logic kept production labor-intensive and limited the designer's ability to achieve a low system price. Technological developments at Intel Corporation after 1971 were about to surmount this last obstacle to the affordable small computer.

1971— :Intel

Intel introduced large-scale integrated memory chips and microprocessors in the early 1970's, providing the basis for the development of low-cost personal microcomputers. The company released

the 4-bit, 4004-based SIM-4 simulator board — the first commercial product incorporating a microprocessor — in May 1972, followed by the 8008-based SIM-8. The Intellec 4 and Intellec 8 development systems, or "blue boxes," followed in August 1973 [6].

The SIM-4 was not a computer, but a design aid to facilitate the development of microprocessor applications. Implementation of the microprocessor in commercial computing really started with the European Micral.

1973: Micral

The French Micral microcomputer developed by REE (Recherches et Études Électroniques) was the earliest personal computer to use a microprocessor [2 & 5]. Truong Trong Thi managed the company and released the computer, which sold for \$1,950, in January 1973. The unit included an Intel 8008 microprocessor and 256 bytes of RAM, expandable to one kilobyte.

France had produced the first microcomputer. Another year would go by before the first North American microcomputer was advertised in the March 1974 issue of *QST*, an amateur radio magazine.

1974-5: Early US Microcomputers

The first personal computer in the USA to use a microprocessor was the SCALBI-8H, designed by Nat Wadsworth and Robert Findley of Scelbi Computer Consulting, Inc. in Milford, CT. Described in advertisements as "The totally new and the very first — Mini-computer," the 8H used the Intel 8008 microprocessor and up to 4K bytes of memory. In kit form it sold for "as low as \$440." The price was right, but Wadsworth had health problems, and a change in the company's first priority — to publishing — resulted in poor sales of the computer.

The first "magazine project" microcomputer, called the Mark-8, was designed by Jonathan A. Titus and appeared in the July 1974 issue of *Radio-Electronics* [10]. It used the Intel 8008 microprocessor and had 256 bytes of memory. Enthusiasts could buy plans from the magazine for a nominal amount, or a substantially complete kit of parts for \$350; sales were insignificant. Today, just over twenty years later, a

Mark-8 is one of the most valuable and sought-after of all historical micros.

It was the misfortune of both the SCELBI-8H and the Mark-8 to stumble and fall in the darkness just before the dawn. Only a few months later, a New Mexico-based company called MITS — Micro Instrumentation and Telemetry Systems — could not cope with the orders it received when it introduced the Altair microcomputer.

1975: Altair 8800

The January 1975 issue of *Popular Electronics* [8] featured the Altair 8800 developed by Ed Roberts of MITS — although the unit on the magazine's cover was only a painted, empty case. It used the Intel 8080 microprocessor, and basic memory was only 256 bytes. A 100-pin bus with 16 slots facilitated expansion of the system with additional memory and peripherals. Kit price by mail order was \$397. The Altair 8800 was a startling success, and a flood of orders created severe delivery problems at MITS; in theory, the same computer was available assembled and tested for \$621, but customers who ordered a finished unit waited for months while the factory caught up.

In retrospect, the Altair's position in the vanguard of the microcomputer revolution is puzzling. The layout of the hundred-pin Altair bus (more commonly known today as the "S-100 bus") was deeply compromised and made the design of third-party peripherals unnecessarily difficult. MITS' own add-on products were sometimes poorly engineered, like the 4K dynamic memory boards, or sometimes were announced but never produced. And if the hardware was attractively priced, the software was not; a paper tape of Bill Gates and Paul Allen's "Micro-soft" Altair BASIC interpreter sold for \$150.

Regardless, the combination of the Altair hardware and the Microsoft BASIC interpreter was the first runaway success of the "personal computer" industry. More sophisticated products with better promotion, like the Apple II computer introduced in June 1977 and IBM's PC released in August 1981, gave substantial computing power to the consumer in the mass market. Which of these early computers we can honor as "the First" will depend on our narrow and literal, or broad and inclusive, interpretation of the term "personal computer" —

which has obviously meant many things to many people over the past forty years!

CONCLUSION

The IBM 610 Auto-Point Computer was the earliest personal computer, but did not use the stored-program concept. The MIT LINC used the stored-program concept, but was a limited scientific project with no impact in the general market. The Kenbak-1, the first low-cost personal computer, had a very limited interface and enjoyed few sales. Xerox' Alto introduced many of the concepts widely used in computers today, but was a "personal computer" only in a rarefied environment and at prohibitive cost. The French REE Micral was the first personal microcomputer to use a microprocessor. The Scelbi-8H and the Mark-8 were the first US microcomputers, but did not have significant success in the market. The Altair 8800 — with a more powerful microprocessor, an innovative BASIC interpreter and an affordable price — was "the first" personal computer to be a commercial success and start a "personal" technological revolution.

BIBLIOGRAPHY

1. Bashe, Charles J., Johnson, Lyle R., Palmer, John H. and Pugh, Emerson W. 1986. "IBM's Early Computers." Cambridge: MIT Press, 1986, pp. 505-508.
— Describes the Personal Automatic Calculator (PAC) and the IBM 610 Auto-Point Computer.
2. Buchholz, Werner (Editor). 1988. "Was the First Microcomputer Built in France?" *Annals of the History of Computing*, Vol. 10, No. 2, 1988, p. 142.
— Discusses the Micral microcomputer.
3. Clark, W. A. and Molnar, C. E. 1965. "A Description of the LINC." Stacy, Ralph W. and Waxman, Bruce W. (eds.). *Computers in Biomedical Research*. New York: Academic Press, 1965, pp. 35-66.
4. Editor. 1989. "Early Small Computers." *Annals of the History of Computing*, Vol. 11, No. 1, 1989, pp. 53-54.
— Discusses the IBM 610, Kenbak-1 and Micral computers.

5. Juliussen, Egil and Isaacson, Portia. 1987. "Computer Industry Almanac." Dallas, Texas: Computer Industry Almanac, Inc., 1987, pp. 288-289.
— Briefly describes the Kenbak, Micral and Altair computers.
6. Noyce, Robert N. and Hoff, Marcian E. 1981. "A History of Microprocessor Development at Intel." *IEEE MICRO*, February 1981, pp. 8-21.
— Describes the SIM-4 & 8 and Intellec 4 & 8 development boards.
7. Press, Larry. "Is There Such a Thing as a Personal Computer." *Abacus*, Vol. 1, No 2, Winter 1984., pp. 69-71.
— Describes "What really distinguishes personal computers from larger mainframes?"
8. Roberts, H. Edward and Yates, William. 1975. "Altair 8800...The Most Powerful Minicomputer." *Popular Electronics*, January 1975, pp. 33-38.
9. Smith, Douglas K. and Alexander, Robert C. 1988. "Fumbling The Future: How Xerox Invented Then Ignored, The First Personal Computer." New York: William Morrow and Co., Inc., 1988.
— Describes development of the Alto computer.
10. Titus, Jonathan A. 1974. "Computer! — Build This Minicomputer Yourself." *Radio-Electronics*, July 1974, pp. 29-33.
— Describes the Mark-8 microcomputer.
11. Wood, Lamont. 1994. "The Man Who Invented the PC." *American Heritage of Invention & Technology*, Vol. 10, No. 2, Fall 1994, p. 64.
— Describes the Datapoint 2200 "smart" terminal.

CHAC DONATES EARLY MICRO TO NIXDORF MUSEUM

On April 6, in a ceremony at the Santa Clara Marriott Hotel, CHAC presented a Processor Technology SOL-20 to the Heinz Nixdorf Museumsforum für Informationstechnik. Participants were Kip Crosby and Edwin El-Kareh for CHAC, Joachim Wolf for the Nixdorf Museumsforum, and former Proc Tech designer Lee Felsenstein.

The donation began with an inquiry to the Smithsonian History of Technology (SHOT) mailing list by Nixdorf curators Ulf Hashagen and Dr. Karlheinz Wiegmann, explaining that one focus of the new facility would be "the personal computer and its development in the USA [and] the history of pioneers and idealists, who got the development of PCs under way." They were, at that time, looking for an Altair 8800, an IMSAI 8080 and a KIM-1; we were happy to suggest the addition of a SOL-20 which, given its Intel 8080 processor, rugged design, and 1976 introduction, is certainly "pioneers' development." Thanks to the generosity of Al Kossow (see **Acquisitions**) we even had such a computer to contribute.

The SOL will travel to Paderborn, DE, for exhibit in Europe's newest museum of information technology. This complete remodeling of Nixdorf Computer AG's original headquarters building has resulted in 18,000 square meters of floor area for the entire museum, of which roughly half will be exhibit space. The remainder will be devoted to conference rooms, a research institute, library and archive, restaurant and museum store. This project is underwritten by the estate of the late German computer pioneer Heinz Nixdorf, through the Westfalen Foundation; it will be "dedicated to the historical development and present significance of information technologies, and their impact on culture and society."

Sounds like our kind of museum and, frankly, we hope someday to be among the 200,000 visitors that the Nixdorf Museumsforum expects to attract each year. In the meantime, we're happy to send one of California's earliest micros to this brand-new and truly fine European institution.

NEW COMPUTER MUSEUMS UNDERWAY IN BAY AREA

by **Frederic E. Davis**
Founder and President,
San Francisco Computer Museum

The Computer Institute Inc., recently incorporated as a nonprofit organization, has joined with the CHAC to create the Bay Area's next major museum of technology: the San Francisco Computer Museum. As envisioned, this Museum will provide a global center for recording and presenting the scientific, historical, artistic, and cultural aspects of computing. An accompanying institute will offer scholars and researchers the opportunity to extend the bounds of computer technology—a profoundly important stage in the development of human civilization.

The San Francisco Bay Area and its environs offer an ideal setting for this facility. Silicon Valley is the birthplace of commercially successful personal computing and remains the world center for computer technology. The City of San Francisco, proposed as the site for the facility, has been ranked the number one tourist destination worldwide, is a major global business center, and serves as a gateway to Pacific Rim countries that are definitive to the computer industry.

The project has received enthusiastic support from the San Francisco Redevelopment Agency and from San Francisco's Mayor Willie Brown. In announcing his support for the museum project, Mayor Brown said "The time is clearly ripe for such a museum with its related educational opportunities, and San Francisco is clearly the place for it."

Phase One: CIVIC CENTER MUSEUM

Long-term plans are to develop the project through several phases. The first phase, scheduled to start within the next year pending final approval from the City, will establish a museum facility in Brooks Hall in San Francisco's Civic Center Plaza. Brooks Hall is a 90,000 square foot facility and former convention site that boasts a notable computer industry heritage; it was the setting for several significant events in computing history, including Doug Englebart's seminal presentation of the

mouse and graphical user interface in 1968, and the West Coast Computer Faire, where the Apple II and many other major computers were launched in the late 1970s.

Based on the theme, "Computing: Yesterday, Today, and Tomorrow," this first phase of the San Francisco Computer Museum will feature exhibits showcasing computing through history, hands-on displays of state-of-the-art technology, and exhibits of the world's premiere computer graphics and multimedia art. It will be the first museum of computing in the world to have such a comprehensive focus.

Later Phases: MUSEUMS, THEATER, INSTITUTE

Later phases of the project have the goal of creating a larger "World's Fair-style" exhibition that will become a major tourist attraction for San Francisco. A major facility is planned that will contain four distinct museums — The Computer Pavilion, The Computer Jungle, The Computer Gallery, and The Virtual Museum — as well as a multimedia theater and auditorium, and the Computer Institute, an educational, cultural, and research resource for both the general public and the computer industry.

THE COMPUTER PAVILION

The Computer Pavilion houses a general exhibit area and a number of company-sponsored pavilions. The general exhibit area traces the history of computing and displays, where one can see early computer artifacts and memorabilia. The company-sponsored pavilions provide a look into the future of computing and are modeled after "World's Fair" exhibits.

THE COMPUTER JUNGLE

The Computer Jungle, less structured than the Pavilion, will include an informal exhibit area offering visitors hands-on access to a wide spectrum of computer technology. Cutting-edge products, and those still under development, will be showcased in exhibits underwritten by the companies responsible for the technology. This will be a cross between an "Exploratorium™ of computers" and an ongoing computer trade show, without the sales hype.

THE COMPUTER GALLERY

The Computer Gallery will exhibit computer and multimedia art. Computers are enabling new artistic expressions that are not easy to display in a traditional art gallery. This will be one of the only facilities worldwide with the technological horsepower to present a broad spectrum of computer art — from multimedia masterpieces to computer animations and virtual reality simulations.

THE VIRTUAL MUSEUM

The Virtual Museum will extend the Computer Museum into cyberspace, through the Internet and other online services, and will also offer special "exhibits" on CD-ROM and other deliverable media. Providing worldwide access to the Computer Museum, The Virtual Museum will also serve as a "community center" without borders, bringing together computer users, scientists, artists, vendors, and enthusiasts in real time.

THEATER AND STORE

The San Francisco Computer Museum complex will contain a multimedia theater and auditorium and an adjacent museum store. The multimedia theater will support live presentation of new works by multimedia artists, as well as previews of new software and hardware, and other visually intensive events difficult to present in a standard theater. Used as an auditorium, the theater will sustain an ongoing schedule of cultural and scientific presentations by scientists, artists, writers, and others. The museum store will offer a selection of computer books, software, art, and high-tech souvenirs to help generate operating revenue for the facility.

THE COMPUTER INSTITUTE

The Computer Institute, which will operate and administer the San Francisco Computer Museum facility, will also sponsor resident fellows in the arts and sciences, who will pursue research into multimedia and computer-interface design. The Institute, unlike research labs operated by private companies, will place in the public domain all software, research, and design work produced by it and its fellows.

COMMPUTERSEUM

Grand Opening Season
July 6 — August 19

by Kevin Stumpf
 Curator, Commercial Computing Museum

Fact: because of 18 people Canada can now boast it has a computer museum.

I am so proud and happy to write an actual review of the Commercial Computing Museum's first exhibition season. I mean it actually opened and people actually came and those people actually enjoyed themselves.

The exhibits were built around mainframe systems. The theme was "input- process-output." The slogan was "A Remarkably Small Museum For Remarkably Big Computers."

Despite the venue and the lack of advertising (because of the lack of a budget), many people (adults, youth, and families) took the time to visit. People came from all over. Many people came from the Toronto and Hamilton areas (about a 60 minute drive), and we were gratified when people from New Jersey and New York arrived. Of course the locals were very supportive, but we didn't see as many teens as we had hoped. Attendance isn't the only indicator of the success of such an endeavor, because when visitors spend several hours browsing, investigating, photographing, and discussing the exhibits something must have been done properly.

The Waterloo County Board of Education provided the venue. We operated from July 6 until August 19 from the gymnasium of University Heights Secondary School in Waterloo, Ontario. This was an unusual place to operate a museum, and no doubt some people didn't take us seriously and refused to visit.

The Board provided the facility rent-free since our mandates are both educational in nature. I think tax payers of the Regional Municipality of Waterloo should be pleased that their board of education acted in such a responsible and creative way to enable Canada's first computer museum to open.

Unlike other attempts at opening a computer museum in Canada, we first obtained artifacts and then opened. Nothing would have happened though without the efforts of many dedicated vol-

unteers. No matter how many ancient computers sat in storage, a group of energetic, helpful, and interested volunteers were needed to wash them, prepare them for presentation, move them, and assist visitors. The Commercial Computing Museum is blessed with such people.

A much more complete review will be published at the end of the year. Until then please visit our site from time to time and stay in touch. Next year's exhibition is already in the works and there are many interesting events and fund-raisers on the way.

We also had a booth at Comdex Canada exhibiting "The Art, Science, and History of Computer Control Panels." About 300 people stopped by. This made July a very busy month. While traveling back from Comdex I was interviewed for a CBC radio show that aired on Labor Day.

If you weren't able to visit the COMMPUTER-SEUM you can still "picture yourself" there. Please spend the time to examine the photographs taken during the grand opening (www.sentex.net/~ccmuseum). Send us your comments and suggestions, as well as your mailing address if you want a copy of the newspaper article about the museum. It's an honest review of the exhibit and idea.

Thanks to every member of the CHAC and of the computer-historical community who gave generously of their support and good wishes during this long project!

Book Review:

THE MICROPROCESSOR: A Biography

Michael S. Malone

New York, NY:
TELOS/Springer-Verlag, 1995
333 pages, \$29.95 (cloth)
ISBN 0-387-94342-0

Reviewed by Kip Crosby

In the Preface to this book, Mike Malone says that he called it "A Biography" because "We anthropomorphize a lot of non-human entities in our world....people have long referred to their computers by their central processors....rather than their brands." Well, maybe, but it's hardly necessary to "anthropomorphize" the microprocessor to recognize the trait it most shares with humanity — a rapacious and half-blind destiny to flow into any niche that might accommodate it, nudging any number of applecarts into chaos on the way. I might once have written this review on a typewriter, but instead I'm in front of a computer so freighted with microprocessors that I couldn't count them all. The computer would be bored just listening to me type, so it's playing a CD on a drive whose laser is held steady by a....you guessed it. And I'm drinking coffee that I just heated up with microwaves whose intensity was controlled by a....uh-huh. The microprocessor is a fit subject for a biographer because it is *pervasive* — and invasive and transformative — in degrees almost appropriate to a living thing.

Now, of course, microprocessors have only existed for twenty-five years. "What?" "Huh?" "No way!" Okay, you're right — not quite yet, because the silver anniversary of the Intel 4004 won't happen till November. Yet microprocessors are now so ubiquitous that *you* — whoever you are — don't know how many you own. Even if you count the ones you can think of (scholastic exercise to say the least,) you'll miss the ones you don't know about. They're called "embedded controllers" and you have more of them than you can imagine. How did we get here — and in a mere quarter of a century? And that question, in its startling and sometimes bewildering complexity, is the one that Malone attempts to answer in a remarkably entertaining book.

Like any good biography, this one requires unrelenting craft in its writing, since years of life are condensed into a few hours' reading (talk about lossy compression!) that still has to cover the high points, tell an understandable story, pay homage to the passage of time, and leave us with a coherent image of the subject. But by taking the storyteller's approach to the microprocessor, Malone gives himself a hard row to hoe, because most biographers can rely on a certain baseline of community that unites subject and reader. Not so this time, and Malone has to begin by telling us what a microprocessor is, where it originated, how it's produced, how it works and how it fails. Imagine for comparison the biography of an extraterrestrial, which would have to include long sections on the creature's anatomy and internal medicine, its languages, and the energy cycles and atmosphere of its home planet. (The atmosphere of a microprocessor's "home planet" — the wafer fab — is not much like what we breathe, being as dust-free as humanly possible. Soon, Malone assures us, even air will not suffice as a medium and fabrication will have to be done in vacuum.) So this book already comprises a small library.

Groundwork thus laid, it's time for the microprocessor's definitively tangled history, with its elements of almost Shakespearean treachery (the disintegration of Shockley Semiconductor and founding of Fairchild was only a spectacular overture,) and its names worse than Tolstoy's — can you tell Xilinx, Xidex and Zilog apart, in a hurry? Only a Silicon Valley native could easily navigate all this, but Malone traces the industry's incestuous connections with such agility that we come to understand a great sea-change; the microprocessor, changing everything, more than anything changed *its own industry*, combining irreplaceable minds into temporary companies, letting designers hop from tilt-up to tilt-up while they amassed experience that earned fortunes again and again. One of this book's finest explorations is of intellectual property in its full flaming weirdness, which has finally turned litigation from a weapon into a tedious kind of insurance.

These changes, once begun, are not spent; in fact, they probably haven't reached full acceleration. At the beginning of this book I grumbled when Malone called the Intel 8080 the "invention of the century;" by the end I realized that microprocessors are practically inventing themselves, and the

century has four years to go. In the book's last section, "Dreams of Light," Malone turns his clear and quick style of inquiry to the concepts and technologies — like fuzzy logic, neural networks, optical circuits, and replacements for silicon — that will drive forward the next consolidation of the microprocessor revolution. As much as we've seen, we await wonders that will stun us to silence again.

These few paragraphs are only a sketch of a book that might as well be called "The Microprocessor: An Archaeology, Anatomy, History, Ecology, Assessment and Prognosis." It's not a completely even book, because the range of topics is an awkward fit in a single binding. The author's talent for explanation, honed by a long and conscientious career as a journalist, is tested to its limits by the deep wizardry of technical fine points. And if the illustrations are both excellent and necessary, the book's overall design depends on contrasts that are sometimes distracting. Still, I don't mean to be too loud about a few little kinks. Building adroitly on the insider's perspective of Malone's first book, *The Big Score*, this book is ambitious, energetic, and compelling; it tries to give a real, rigorous picture of historical transformation still in wild process. *The Microprocessor: A Biography* will appeal to any reader who wants to understand integrated electronics as primal force.

ACQUISITIONS

From smoldering rumor to blazing truth — yes, the CHAC is accepting hardware again. And hardware for the ages! We need to know more about a lot of these computers (and take a longer look at some of them) but here's a barebone description, for the record, of our accessions in the last quarter. Share the joy, o ye faithful. The ENGINE's next allocation of time and space will bring pictures and detailed stories of many of these machines.

Apple Lisa 2 and Apple Mac XL; thanks to Craig deRosa. These aren't quite from the computer store. One of them has a slotted, transparent plexiglass front panel — we suspect made in a prototyping shop, not hacked — and a ten-meg internal hard disk....

Compupro homebrew; thanks to James Birdsall. A Z80-based S-100 in a rack cleanly fabricated from plywood and sheet aluminum. Lots of spare boards including a Hayes modem; lots of software; docs on *everything*, neatly alphabetized in manila folders. The definitive museum-piece micro.

Data General Nova II; thanks to Max Anthony, and we'll have details later.

GRID Gridcase 3; thanks to Mike Tassano. The secret agent's laptop, in stealth-black cast alloy. The orange plasma screen is so sharp that the pixels are visibly square. Honestly a bit heavy by modern standards, but it could double as body armor in a pinch, and it looks so....*serious*.

HP 2114A; thanks to Cliff Olson. One of the old, old, original-original HP rack-mount minis, with a single-digit prototype serial number. Beautiful in that faintly patinated way that brings to mind, say, a classic car. Stay tuned for a feature.

HP terminal and tape drive; thanks to Max Anthony, and we'll have details later.

Intel MDS; thanks to the Tech Museum of Innovation. This is one of the later Intel development "blue boxes," with an integral CRT and (we suspect) an i8086, rather than an Intellec. It came with what look like full docs for Isis.

Intertec Superbrain; thanks to the Tech Museum of Innovation. An early attempt at an all-in-one business micro, this combined computer/terminal boasts dual Z80's, dual 8" floppies, a 12" CRT and 32K RAM. (Note to younger readers: In 1979 that was a lot.) The whole package is clean, imposing and humongously heavy. Not a bad computer by the standards of the day, the Superbrain has faded so far back into history that some people today think it's a game machine.

Kaypro II; thanks to Craig deRosa. A second example of the well-loved blue-and-silver luggable, this — like all spares — will make the rounds as an exhibit computer. Although we haven't yet checked the two computers side by side for minor distinctions....but then, were any two K-twos ever totally alike?

Osborne One; thanks to Mike Tassano. A pristine and "late," gray-cased example of the blockbuster Z80 luggable that every laptop and notebook owes its soul to. Only a few years after cozying up to a desktop, micros — thanks to the Ozzie — were free of even that. A true classic joins the roster.

Processor Technology SOL-20; thanks to Al Kossow. Actually, *two* more of the handsome and gutsy pre-Apple micros with walnut sides. With Al's enthusiastic permission, we donated one to the new Nixdorf Museum (see p. 46;) the other will become an exhibit computer.

Seattle Computer Products 8086; thanks to Charlotte D'Amico. A connoisseur's micro on two counts. First, this *may* have been the first production 8086 computer outside Intel. (Article topic! Article topic!) Second, it was one of these — a close relative at least — on which Tim Patterson developed QDOS, the precursor of MS-DOS; and, probably, on which he alpha-tested Microsoft 8086 BASIC in May 1979.

Tektronix 8562; thanks to James Birdsall. This rarely seen article is a rack-mount development box based on a ceramic-pack DEC LSI-11. Built at a time when Tektronix had an almost unrivaled reputation for quality, the 8562 is of achingly beautiful construction. Since the outside is beige and has the usual eight corners, we may display it with the case off. Watch for a feature.

Xerox Alto; thanks to Al Kossow. *Yes!! Really!* Absolutely! The CHAC has an Alto, with its graphical interface, its Smalltalk, its fourteen-inch, 2.5 Mb hard disk.... This computer had stuff *no* other computer had in 1974. This computer has stuff *your* computer doesn't have *now* — like a full-page portrait monitor. And all of this pales next to the fact that it plays Galaxian. To sit and play at the Alto, with Butler Lampson's black-covered manual at elbow, is to enter a *major* time warp.... in which the roots of LisaOS, MacOS, Atari GEM, X-Windows, MS-Windows, etc., are finally laid bare to the dazzled seeker. One of the few computers that can be totally ingratiating and still send chills down your back.

Tech Corner:

DE-CRUFFTING a POWER SUPPLY FAN

by Kip Crosby and Joan Piker

[*Note:* The procedure outlined here, if performed correctly, will clean the power supply generally to be found in a commodity desktop or tower computer. Neither the authors nor this publication are responsible for loss or damage suffered as a result of performing this procedure incorrectly, or performing it on a unit not of the design described below. *Caveat lector.*]

The fan in the power supply of an average desktop computer leads a ghastly life. Expected to deliver consistent throughput and constant uptime, it survives on a diet of spiky wall voltage and dust-laden hot air. The hot air flows through the fan and does its bit for global warming; much of the dust — spongy gray gunk seeded with evil glitter — accumulates on the fan blades or, worse, within the power supply box itself. Wouldn't it be nice, you think, to really get in there and suck out all that cruft? But it's not that easy....

An inexpensive power supply is typically cooled by a muffin fan mounted *inside* the power supply box. The fan is covered by a protective grid of either stamped sheet steel or welded wire. Four bolts, one at each corner of the fan, run through the protective grid, through tubes in the corners of the fan shroud, then through speednut clips that secure them. (Figure 1 shows a fan frame, with the fan omitted for clarity, and one speednut in typical position at the upper left.)

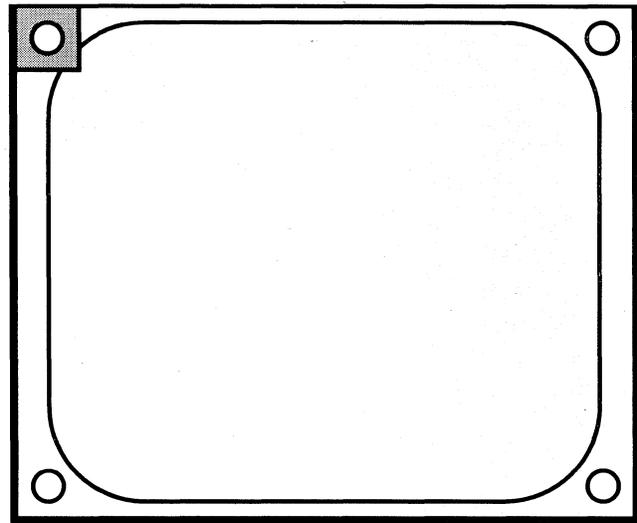


Figure 1.

The dread begins when you realize that this entire sandwich, held together by the bolts, can never be removed from the tack-welded steel case. You have to do all your work inside the case, while you avoid dislodging the speednuts, which would fall in and be almost impossible to retrieve. (And no, you can't leave one in there, or you risk shorting out the power supply circuit board.) This maintenance is a challenge, but it's possible, and with a little extra care, it can even be repeated when necessary.

Tools and materials list:

- Screwdriver, probably #2 Phillips
- Hex driver, probably ¼"
- Mini vacuum cleaner with wand tip
- Bench rag
- Large paper clip or other short stiff wire
- Round toothpicks
- Superglue pen
- Double-face narrow (¼") tape
- Cotton swabs
- Isopropyl alcohol

1.) Power-down your computer and unplug it. If you're really prudent, let it sit for a while so the capacitors can drain. Go use your other computer.

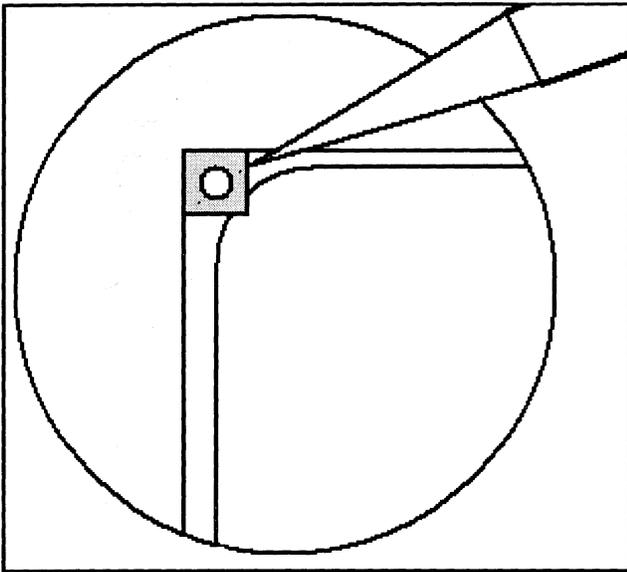


Figure 2.

2.) Spread the bench rag on your workspace. Disconnect the power supply's connectors from the devices and main board, taking note of their orientation as you do. With the hex or Phillips driver, undo and pull the (probably) four bolts that hold the power supply against the backplane of the case — *not* the four bolts at the immediate corners of the fan frame. These will be outside the cutouts for the fan and power cables. In a desktop case, the supply is also probably held in by two tangs at the bottom of the computer case that fit into recesses in the power supply box. Pull the power supply forward — away from the backplane and toward the drive connectors — about three-quarters of an inch (20mm); then lift it up and out, and set it on the rag.

3.) Get all possible dust out of the fan, and off the perforated sections of the box, with the mini-vac. Put the box back on the rag with the fan protector at the top.

4.) Bend the paper clip or wire into a slim hook. Slip it through the fan protector and snag one blade of the fan. Remove the four bolts that hold the fan in place, making sure (as you withdraw the fourth bolt) that you have a good grip on the fan with the wire hook. Be *careful* not to knock the speednuts out of alignment; they are now loose.

5.) Remove the fan protector. With it gone, you can probably get a better grip on the fan with your fingers than with the wire hook. Move the fan around inside the box so you can see each corner

in turn. Carefully clean each corner of the fan frame, and the face of each speednut, with a cotton swab and alcohol.

6.) Make sure each speednut is aligned with the tube that its bolt goes through. Touch the edge of the speednut with the superglue pen (as in figure 2) to glue it to the fan frame. Hold the fan till the glue dries.

7.) Using the minivac, or turning the box upside down and shaking it, remove dust mercilessly. Get the minivac's wand past the fan and into the box, to clean the circuit board gently.

8.) Put a small piece of double-face tape over each speednut. Set the box down with the fan facing the edge of the table, make sure the fan is aligned properly, and pull the fan back into place to stick it to the inside of the box. Poke a toothpick into each bolt hole, through the tape.

9.) Hang the fan protector over the toothpicks. Replace each toothpick, one at a time, with a bolt. Re-connect the power supply's connectors and check any other cables you may have jostled.

You're done! The fan will be significantly more efficient, and probably quieter. When you want to do this again, you can just take out the bolts and unstick the tape, without worrying about the glued-down speednuts.

QUERIES

APPLE ONE

Chris Bachmann, c-bachmann@nwu.edu, is trying to compile a registry of owners of the Apple One — partly to share increasingly scarce information, and partly to find out how many boards still survive. Interested parties can reach Chris via e-mail or care of Bachmann, 225 N. 3rd Street, Wheeling IL 90090 USA.

APPLE LISA 2/10: MYSTERIES

I recently acquired a Lisa 2/10 from my school, running an outdated version of MacWorks. I have a few questions which I have been asking since the purchase of the machine. If anyone here could lend a hand, thanks in advance.

1. Under the CRT, there is a small panel listing the serial number, Applenet #, and Manufactured #. What do all these fields mean?
2. The hard drive makes a high-pitched squeak whenever the arm moves out of and into the parked position (I'm guessing, I know at least that it's the arm). Is this normal? The disk doesn't seem to be suffering, and no data is lost. If this needs to be remedied, what can I do?
3. What is a good way to get that hard-to-reach crud out of various cavities in the body? Would Dust-Off or a similar product be a bad idea?
4. The keyboard is covered with that familiar Apple keyboard crud that accumulates on all keys of Apple keyboards (maybe others). How can I remove this?
5. Out of curiosity, the fact that I have an old Mac, and the fact that the Lisa makes a rotten Mac, I am reinstalling the Lisa Office System. All I really need is LisaTerm (which is a term program, right? I've only seen the Office System once; my apologies if I am using incorrect terms), but does the Lisa support any sort of networking whatsoever?

Gotta go, thunder...

Tom Stepleton
 ssteplet@artsci.wustl.edu

APPLE MACINTOSH

I have recently rescued a Macintosh 128 from a trash heap. I have made what repairs I could, but need some system software to try it out! Please send me any information you have regarding where I might get hold of some. I sure would like to get this little toaster smiling again! Thank you for your time.

Stephen Jones
 swj0001@jove.acs.unt.edu
 UNT
 Box 7519
 Denton TX 76203
 (817) 243-5242

BOSTON IV

I'm looking for the company, or any info, on a Boston IV home computer system. All I know is that it had a 6502 processor. Can you help? Thanks....

Chuck
 chucks@psln1.psln.com

COMPUTER AUTOMATION

Does anyone at CHAC know of a place from which I could purchase programming manuals for certain obsolete minicomputers (without the associated hardware)? My interest is in one of the lesser known manufacturers, Computer Automation, Inc. They were probably forced out of the mini-computer manufacturing business by the advent of the microprocessor, but used to be in Irvine, California. The models I am interested in are the Alpha-16 and the LSI-4 minicomputers. Thanks,

Roy Campbell
 75537.2422@compuserve.com

DATA GENERAL NOVA 3

I recently acquired a Data General Nova 3 mini-computer. Anyone with memories or information on this beast? I'd like to know more of its history.

Is it true that the Nova design was initially rejected by DEC?

Edwin R. Parsons

edwin@peninsula.apana.org.au

EAGLE PC-2: DOCS AND S/W WANTED

Arthur Bauman has recently inherited an Eagle PC-2, an Intel 8088-powered California IBM PC clone that can run either MS-DOS or CP/M-86. He would appreciate hearing from anyone who has docs or software for this machine, especially a user's manual. If you can help, please write to him at 124 Orchard Avenue, Mountain View CA 94043.

80 MICRO MAGS OFFERED

I have a complete set of 80Micro Magazines. The magazine for the Radio Shack TRS-80 Model I, II, III etc. The collection includes Issue 1 (1-80) through Issue 74 (3-86) With a special edition published Jan 1983. Wayne Green was the publisher.

I am looking for a good home for the magazines, please contact me if you would like them or if you know of an organization which would like them.

Regards,

Rick Hoover

rick_h@community.net

ENIAC DOCS?

Does anyone know where I can find technical documentation for the ENIAC computer? I would be very interested in detailed descriptions of the hardware and instruction set. Ultimately, I would like to write a ENIAC simulator for the PC and run some of the original programs if I could get hold of them.

Jim Stewart

jkmicro@dsp.com

IBM

I have a bunch of IBM unit record equipment including a large number of plug boards card sorters with tubes and printers. When it was taken out of service a few years ago it still ran. Any idea of who might be interested in it and how much it would be worth.

Thanks,

Bob Swartz

rs@interaccess.com

IMSAI

Do you know of anyone selling any IMSAI 8080's? I am VERY interested in obtaining one with floppy drives. Being the computer in the movie War Games makes it quite interesting.

Thanks.

Nick

conartis@tiac.net

[8080's are relatively common because they were gutsy and performed well, so people kept them. The going rate for a functional one seems to be about \$300. Be warned that you may or may not want the floppy drives, which earned themselves the name of "Pizza Oven;" you might be just as well off with paper tape.]

MINIX/286

I have a couple of old 286s witch I wish to put to slow redundant tasks. And to do this of course I can not use DOS. I am looking for someone willing to sell/donate/pay-me-to-take a useable copy of Minix (I figure everyone must be running Linux primarily nowadays) that would run on a 286 machine.....

E-mail if you can help.

T. Camp

camp@industrial.com

NCR I-9020

I have acquired an old (1978) NCR I-9020 rack mount computer, with ribbon printer, and 20 14inch floppies.

All the hardware seems to be OK, but the SYSGEN tapes are corrupt. I have been unable to locate a source for the tape which is required to boot up the system.

Hope someone has an idea or suggestion. Any help would be appreciated. Thank you,

Joe Mohnike
jmohnike@valley.net.com

OSBORNE ONE: DISKS, DOCS, MAGS

I'm looking for Osborne 1 disks, and any of the Osborne 1 magazines that still might be out there.

Can anyone help? Thanks,

Bill Johnson
bjohnson@moa.com
P. O. Box 579781
Modesto CA 95357 USA

: DISKS, DOCS, or OFFER

I have an Osborne I in mint condition; looking for software and manuals.

Alternatively, open to some sort of counter offer.

Susan Rosen

RRosen@postoffice.worldnet.att.net

**SHUGART 14" DRIVE INTERFACE:
ANYTHING?**

Does anyone have any information about the history of the interface used with the Shugart SA4000 14" hard drives, and information about any other drives that were compatible with this interface?

Also, did the SA1000 interface, which, if I recall correctly, is the predecessor to, and (about?) the same as, the ST412/ST506 interface, precede the SA4000 series hard drives? Can anyone provide any information on the SA1000 hard drives?

I'm trying to figure out if the SA4000 interface was meant to be an improvement upon the SA1000 interface, as it has a higher data transfer rate (7.11

MBit/s vs. 5 MBit/s), and in that it uses NRZ (non-return-to-zero) pulses for the data, bears some resemblance to ESDI (although 10 to 15 MBit/s, it also uses NRZ for the data, and uses MFM encoding, like the SA4000 and SA1000/ST506).

Also, out of curiosity, are there any others out there, aside from those of us still using PERQ-1 graphics workstations, who are still using these 14" Shugart hard drives (or remember using them)? I'm sure that there must (hopefully) be some people out there who are still using these drives with Z80, S-100 bus, based CP/M systems as well, as there was a Discus M26 Winchester Disk System controller used with these S-100 based systems. The User's Manual for the Discus M26 (written by George Morrow) is rather nice, containing schematics and source code. Is anyone still using the M26?

Thanks very much in advance for any information that anyone can provide about any of the above!

R. D. Davis
<http://www.access.digex.net/~rdd>

WANG 320SE

I have a ~1967 Wang Model 320SE multi-user electronic calculator unit. It is approximately 60x20x13cm and weighs about 15kg. It has four connectors for external keyboard-display units. I would like to find more information about this beast. Do you know where I might look? Any information would be welcome.

Thanks,
Scott Coburn
scott@bnl.gov

XENIX

Hey, HELP. I am looking for a copy of Xenix-286 Ver. 3.2 (2.3). I need the boot disk and utility disks. I am surfen' the net for any help I can get. Thanks

Dennis Rapp
rapp@i-link.net

ZILOG Z8000: ONLINE DOCS?

Does anyone know where to find some online documentation on the Z8000 family of CPUs? I wrote email to someone that said Olivetti made a computer based on one of the Z8000s, does anyone out there know what it's called? I'm also trying to track down a cross-assembler for the Z8000 (the Z8002, specifically, which can only address 64K), and I think there's a commercial table assembler available that has the Z8000 in it, but it's \$200 and I was hoping to find a freeware or shareware program that might have it. Any info would be greatly appreciated!

Gary
garyd@haus.efn.org

PUBLICATIONS RECEIVED

In the interest of getting ENGINE 3.3 out the door during the present century, listing of publications is postponed until the archive ("heap") can be reviewed and reorganized. We hope to finish an updated list in time for 3.4.

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 (formerly Unusual Systems,) 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, 300 Congress Street, Boston MA 02210. No contact at present; use www.tcm.org.

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

Computer Preservation Society (Inc.), Ferrymead Historic Park, 269 Bridle Path Road, Christchurch 8002, New Zealand. Abraham Orchard, secretary; abe@voyager.co.nz.

Computer Technology Archive, Box 4376, Stanford CA 94309. Bill vanCleemput, director.

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.

International Association of Calculator Collectors, 14561 Livingston Street, Tustin CA 92680-2618. 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 Kojane, president.

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

THANKS TO....

Aaron Alpar for his donation.

Charlotte D'Amico and Dan Lahey for hand-delivering the Seattle Computer Products 8086.

Fred Davis for the Dell Dimension 486-100 now serving as our full-time fax machine, among other things.

George Durfey for his help with the Maiga Wolf interview.

Andrew Eisner for the Macintosh SE/30; Frank Freeman for the SE video board; and Gavin Carothers for setting it all up with System 7.1.

Tom Ellis for ongoing support of MS-FrontPage and MS-Windows NT; and for configuration of our brand-new Ethernet network (*yesssss!!*)

Edwin Vivian El-Kareh for his formidable contributions as your Association's Tactical Director.

Don Kojane for his collaboration and support on the lecture circuit.

Bob Ragen for fact-checking Nick Bodley's Friden article.

Erich Schienke for fine-tuning this issue's graphics.

NEXT ISSUE / COVER ART

To begin with, it's the silver anniversary of the microprocessor and *that's* worth commemorating....there's an interview in the works with a workstation pioneer.... maybe an excerpt from a major new computer history title....and a review of Hafner and Lyon's new book on the origins of the Internet. In your hands as soon as we can get it there.

Cover: Maiga Wolf at work in London, Ontario, 1949.

GUIDELINES FOR DISTRIBUTION

The ANALYTICAL ENGINE is intellectual shareware. Distribution of *complete, verbatim* copies through online posting, Internet mail or news, fax, postal service or photocopying is encouraged by the Computer History Association of California.

Excerpting or brief quotation for fair use, including review or example, is also permitted, with one exception: Any material copyright to or by a third party and reprinted in the ANALYTICAL ENGINE by permission shall not be used in another periodical or context, unless the permission of the copyright holder is separately secured for the new use.

Alterations, abridgments or hacks of the ANALYTICAL ENGINE which change the intent or meaning of original content; or which contrive to bring income to any person or organization other than the Computer History Association of California; or which contrive to injure the Computer History Association of California, its officers, contributors, volunteers or members; are PROHIBITED. Reproduction of the ANALYTICAL ENGINE without its subscription coupon is abridgment in this sense.

GUIDELINES FOR SUBMISSION

The ANALYTICAL ENGINE solicits manuscripts of 750 to 2500 words on the general topic of the history of computing in, or with significant reference to, the State of California. Articles should focus on one interesting or illuminating episode and should be written for a technically literate general audience. Submissions are welcome from both members and non-members of the CHAC. Article deadlines are: July 15 for the November issue, October 15 for the February issue, January 15 for the May issue, and April 15 for the August issue.

Each author may publish a maximum of one signed article per year. This restriction does not apply to letters, queries, book reviews or interviews. Thank you for cooperating to protect diversity of voices and topics. Previously published material will be republished only in clearly attributed

quotations or citations; or when its publication in the ANALYTICAL ENGINE will bring it to the attention of a significantly broader audience; or when the original publication is materially obsolete or inaccessible.

Decision of the editors is final but copyright of all published material will remain with the author.

The preferred document file format is Microsoft Word for DOS or Windows, but almost any DOS or Macintosh word processor file will be acceptable. Submit manuscripts on DOS 5.25" or 3.5", or Mac HD (1.4) diskettes. Alternatively, please send your article as ASCII or ISO Internet mail. Please avoid submitting on paper unless absolutely necessary.

The ANALYTICAL ENGINE

Volume 3, Number 3, May 1996

ISSN 1071-6351

newsletter of the Computer History Association of California, is published four times a year — in February, May, August and November — at Palo Alto, California.

This magazine is available both on-line and on paper. Basic, domestic subscriptions are \$25 electronic and \$35 paper, with \$25 deductible as a charitable donation. For information on institutional, international, and low-income subscriptions, contact the Association at:

**4159-C El Camino Way
Palo Alto, CA 94306-4010 USA**

Internet: engine@chac.org

WWW: <http://www.chac.org/>

NINES-CARD

WHO INVENTED "SOFTWARE?"

by Herb Kanner

In 1958, I managed by sheer *chutzpah* to wangle myself an assistant professorship at the Institute for Computer Research, University of Chicago. My graduate degree was in physics, and all that I knew about computers was that I had written some programs for an IBM 650 in assembly language, in Perlis's IT, which was perhaps the first high-level language on earth, and in the Bell Labs interpreter, which simulated a three-address machine.

The Institute was building a transistorized computer (one of the first, if not THE first) on an Atomic Energy Commission contract. So here were these guys soldering transistors, capacitors, and resistors onto printed circuit boards and quaintly referring to the stuff as "hardware." This I found very amusing; my concept of hardware was something you bought in a hardware store.

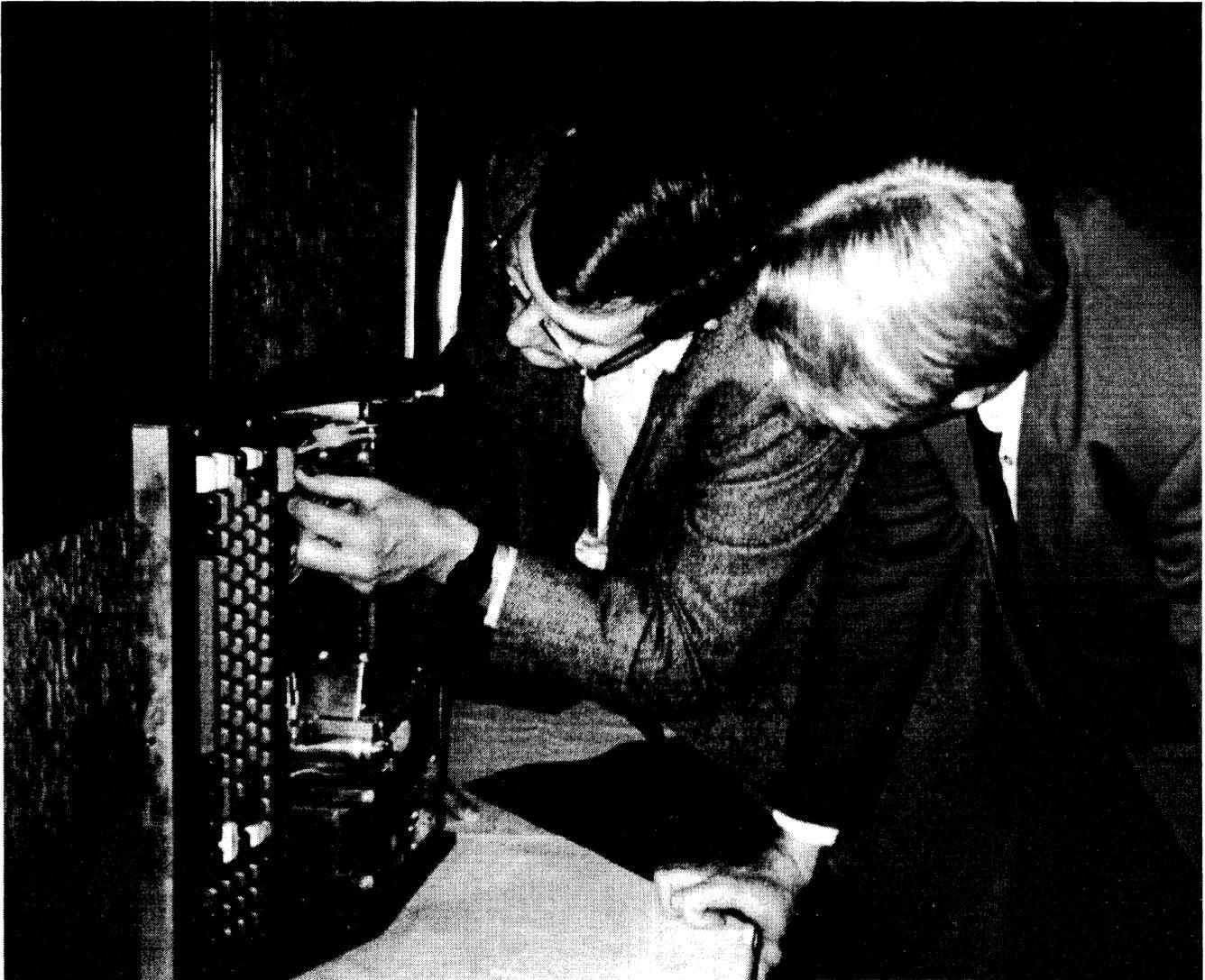
Because I was responsible for the initial programs for this machine, e.g. a symbolic assembler, I thought it would be a cute idea to put a sign on my office door that read "Software Department."

I'm sure that the term "software" did not radiate from my usage. But I'm also fairly sure that it was not floating around the literature at that time. So, at worst, I made an independent invention. It might be interesting to try to track down the other origins of the term.

CONTENTS

Editorial: THREE YEARS	1
IN MEMORIAM: DAVID PACKARD	3
THE COMPUTER WILL DO ANYTHING YOU TELL IT TO DO: An Interview with Maiga Wolf by Hilary Crosby	5
Quick Take: RAMAC 40 th ANNIVERSARY!	20
Quick Take: GETTING FRAMED IN 3D	20
THE MAC AND ME: 15 Years of Life with the Macintosh (Part 2) by Jef Raskin	21
SPEAKING OF ENGINES.... ..	34
FTP SITE FOR COMPUTER CONSERVATION SOCIETY, by Chris P. Burton	34
THE FRIDEN EC-130: The World's Second Electronic Desktop Calculator, by Nicholas Bodley	34
What Was The First PERSONAL COMPUTER? <i>An Exploration</i> , by Roy Allan	42
CHAC DONATES EARLY MICRO TO NIXDORF MUSEUM	46
NEW COMPUTER MUSEUMS UNDERWAY IN BAY AREA, by Frederic E. Davis	47
COMMPUTERSEUM Grand Opening Season July 6 — August 19, by Kevin Stumpf	48
Book Review: THE MICROPROCESSOR: A Biography, by Michael S. Malone	49
ACQUISITIONS	51
Tech Corner: DE-CRUFING a POWER SUPPLY FAN, by Kip Crosby and Joan Piker	52
QUERIES.....	54
PUBLICATIONS RECEIVED	57
ADDRESSES OF CORRESPONDING ORGANIZATIONS	57
THANKS TO.....	58
NEXT ISSUE / COVER ART	58
GUIDELINES FOR DISTRIBUTION	58
GUIDELINES FOR SUBMISSION	58
NINES-CARD: WHO INVENTED "SOFTWARE?," by Herb Kanner.....	59

SUBSCRIBE!



Lee Felsenstein and Joachim Wolf examine the SOL-20 (page 47)

and....

The Computer That Could Do Anything • David Packard, 1912-1996
The Earliest PC's • Friden Calculators • Raskin's Mac and Me, part 2
Malone on Microprocessors • San Francisco Computer Museum
Dealing with Fan Cruft • Nixdorf Museum in Paderborn • and more!

The Computer History Association of California