

BONUS: BYTE Guide to Apple PCs

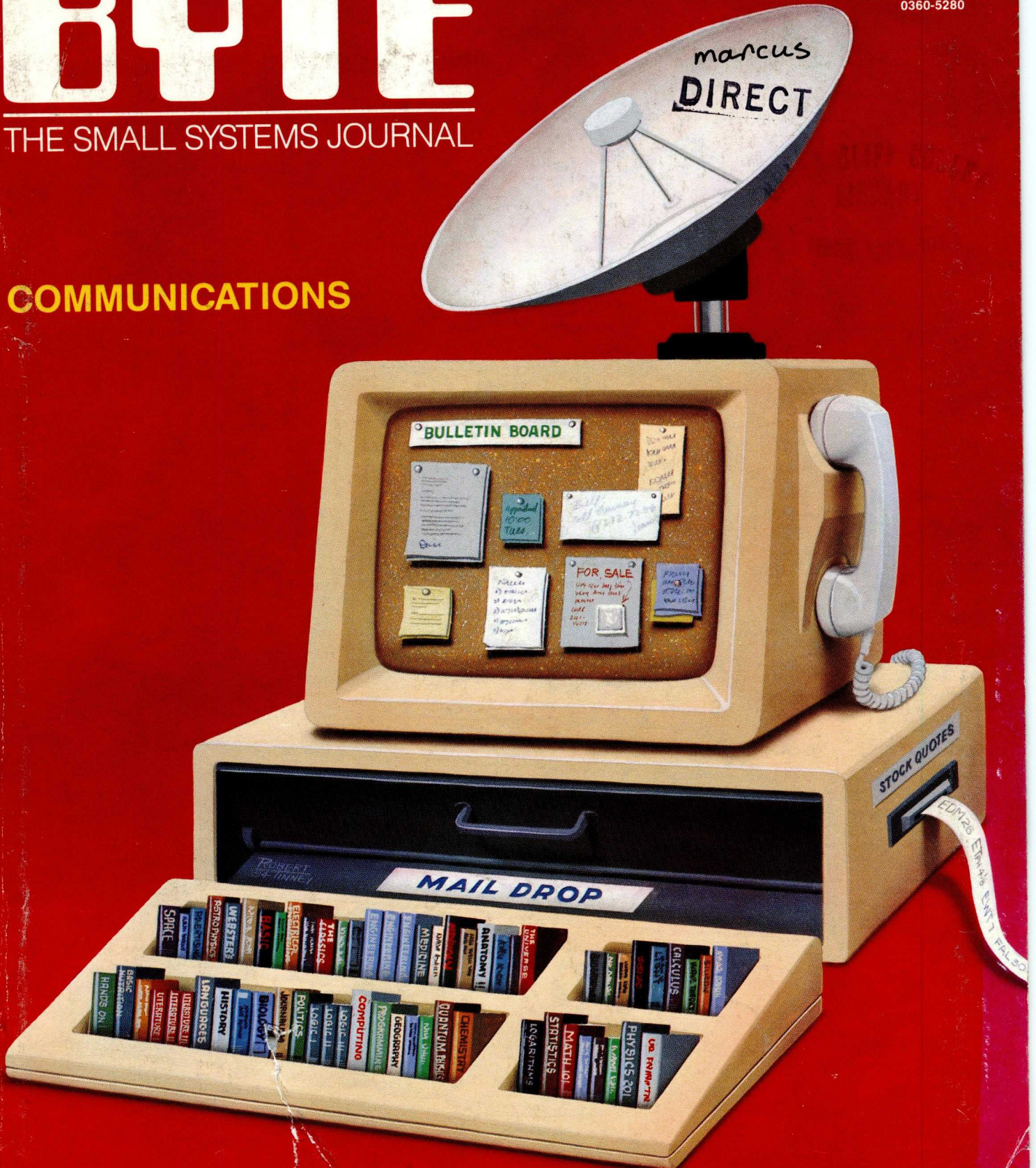
BYTE

THE SMALL SYSTEMS JOURNAL

DECEMBER 1984 VOL. 9, NO. 13


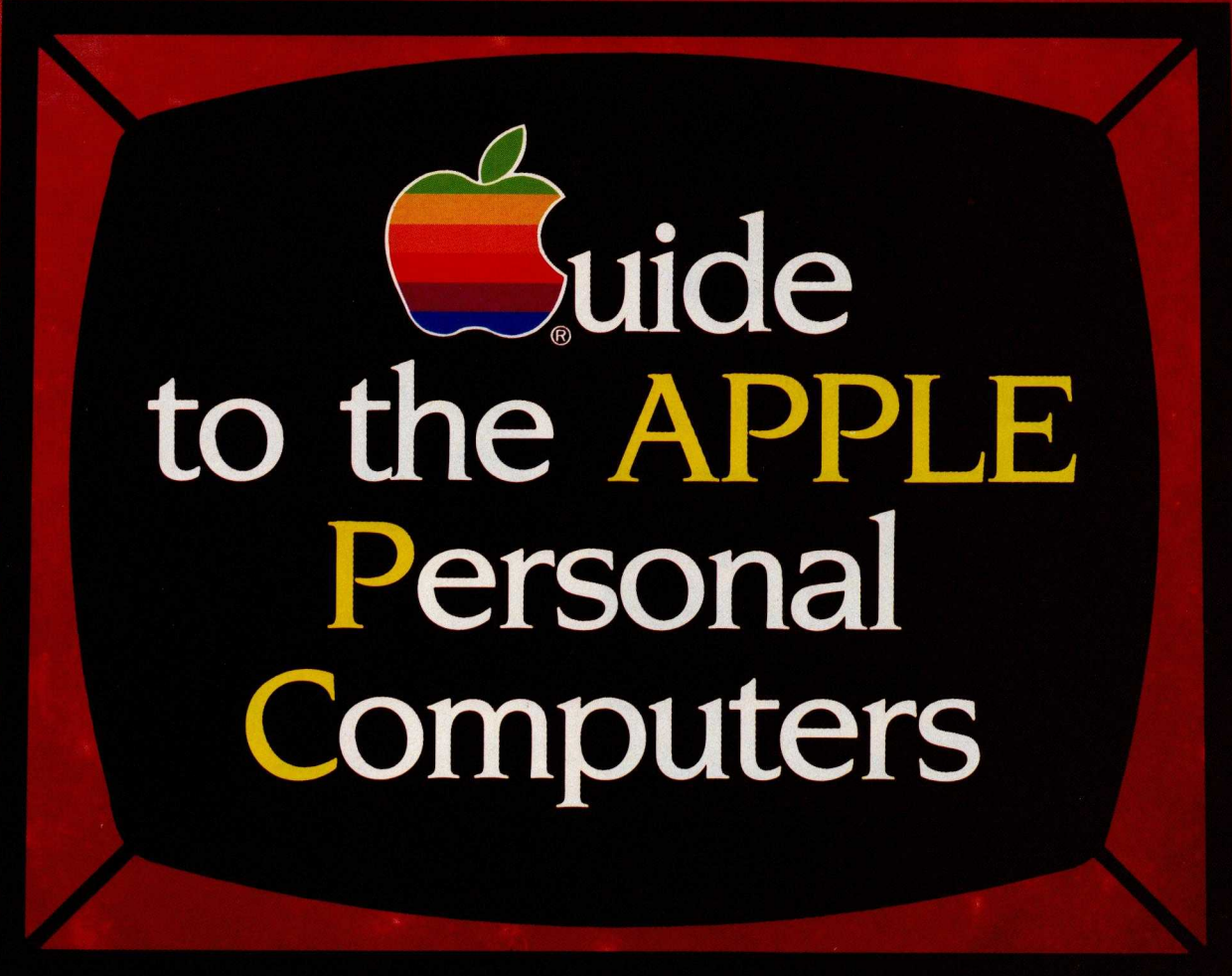
\$3.50 IN UNITED STATES
\$4.95 in CANADA / £2.10 in U.K.
A MCGRAW-HILL PUBLICATION
0360-5280

COMMUNICATIONS



BYTE

THE SMALL SYSTEMS JOURNAL



uide
to the **APPLE**
Personal
Computers

Apple IIe

Apple IIc

Macintosh

Lisa 2

SoftCard squeezes the most juice out of your Apple.

CP/M

Microsoft® Premium SoftCard™ IIe is the high-performance CP/M® board that really juices the Apple® IIe.

Hard facts on SoftCard.

It has a high speed (6MHz) Z-80 that runs CP/M up to three times faster than lesser boards. Plus 64K memory and 80-column display that fits the IIe auxiliary slot and acts like Apple's own Extended 80-column Card. So it works with CP/M, Apple DOS and ProDOS programs, too.

Microsoft BASIC is built-in, so it's compatible with more Apple CP/M software than any other board on the market: Thousands of the juiciest business programs including dBase II®, WordStar® and sophisticated Microsoft languages like

FORTRAN-80, COBOL and BASIC Compiler.

It also has a new low price.

Juicing up the performance of computers is nothing new for us. We invented the SoftCard and make versions for the entire Apple family. We wrote Applesoft for the Apple II.

MICROSOFT In fact, our The High Performance Software® BASIC is the language spoken by nine out of ten microcomputers worldwide.

Get the Apple juicer from Washington. Call 800-426-9400 (in Washington State call 206-828-8088) for the name of your nearest Microsoft dealer.

SoftCard is a trademark and Microsoft is a registered trademark of Microsoft Corporation. Apple is a registered trademark of Apple Computer, Inc. IBM is a registered trademark of International Business Machines Corporation. dBASE II is a registered trademark of Ashton-Tate. WordStar is a registered trademark of MicroPro. CP/M is a registered trademark of Digital Research, Inc.



**MICRO
SOFT**

THE Part 1: Early History APPLE STORY

CONDUCTED BY GREGG WILLIAMS AND ROB MOORE

An interview with Steve Wozniak

Steve Wozniak is the designer of the original Apple computer and one of the cofounders of Apple Computer Inc. Here Steve speaks at length on a variety of topics that range from the hobby activities that led to his design of the Apple I to current issues at Apple Computer. His frank comments and criticisms provide a glimpse into the workings of Apple Computer from a true insider's point of view. This is the first part of a two-part article.

BYTE: *We've heard that you designed the Apple I while you were working at Hewlett-Packard in 1975. Can you tell us what you were doing before that; what led up to the creation of the Apple I?*

Wozniak: I have been interested in electronics and computers all my life. In my high school days, I studied TV circuits and I designed about 50 computers on paper, but I couldn't afford the parts to build them. In 1970, most people couldn't afford a monitor, so I designed video outputs that plugged into an oscilloscope and had the oscilloscope draw letters on the screen.

I had also been studying software. I wanted to know how to write compilers for languages like FORTRAN or BASIC, so I studied and kept notebooks. It was all self-taught and on paper, but I never got a chance to try it out.

I took three years of college toward a computer science degree back when only a few colleges were starting to offer it as an undergraduate curriculum. My third year had been at The University of California at Berkeley.

I originally planned to take a year off from Berkeley after my third year to earn enough money as a technician for my final year of college. But my career kept advancing. I was hired by Hewlett-Packard, made an engineer there, and I started developing better design expertise. I got into chip layout and

things like that, and basically my career kept moving, my interests in life were changing, and it was too far to commute to Berkeley. I tried San José State but I didn't have the time available and it wasn't compatible with my first three years of college. It would have taken four more years to get my degree. So I never got a degree.

I was still an electronic hobbyist. I lost interest in minicomputers during the next three years, because I was doing calculator chips at HP and projects on the side at night.

I saw a Pong game in a bowling alley. So I went and designed my own. I designed the Breakout video game for Atari. I was just constantly involved in electronics as a hobby. At Hewlett-Packard we were pretty much just designing integrated circuits.

Around that time the Homebrew Computer Club got started and I happened to get directed to the first meeting by accident. I started discovering a bunch of high school kids who knew all about microprocessors and assembly language, and it was the stuff I had done up until three years before in my life. My whole life had been in minicomputers. All of a sudden, I started realizing microcomputers are the same as minicomputers, and I understood them.

BYTE: *You mentioned that you designed the Breakout game for Atari. How did that happen?*

Wozniak: Steve Jobs was working for Atari at the time. Nolan Bushnell was really annoyed because all their new games

(continued)

Gregg Williams is a senior technical editor at BYTE. Rob Moore is a hardware designer and a frequent contributor to BYTE. They can be reached at POB 372, Hancock, NH 03449.

'I didn't want to pay to use somebody else's computer, so I decided to design my own.'

were coming out at 150–170 chips. He wanted low chip counts to reduce costs, and he had seen a version of Pong that I had done, that only used about 30 chips. He appreciated that. So he said if we could design a hardware Breakout in under 50 chips, we'd get 700 bucks; and if it was under 40 chips, we'd get \$1000.

Atari didn't put us on a time schedule; Steve did. I had to do it in four days because Steve had to catch an airplane to Oregon. I was the designer—the engineer—and Steve was a breadboarder and test technician.

We gave them a working breadboard for it. My first design was 42 chips. By the time we got it working it was 44, but we were so tired we couldn't cut it down. So we only got 700 bucks for it.

THE APPLE I

BYTE: *How did you get to the point of designing the Apple I?*

Wozniak: I had worked my way up through software using a terminal on a local time-sharing system. Sometimes I'd call it from work, but I wanted to do it at home. I eventually designed and built a TV terminal and a modem so I could call this computer and play games. I was a true hacker. I started getting on ARPA-net and accessing computers all around the country. BYTE was the first magazine to get started, and I bought issue #1 at the newsstand.

I didn't want to pay to use somebody else's computer, so I decided to design my own. I wanted to have it all in one place, and I already had a terminal, so I was part-way there.

I sat down and wrote the BASIC first, and that took a lot more time than the computer design. Once it was written, I had to build the computer. I had designed a simple 6800-based computer, but the actual choice of processors was dictated by what I could afford. At the time, most micro-processors cost hundreds of dollars, but you could buy a 6501 over the counter for \$20, and a 6502 was \$25. So I bought a 6502, built the computer, and soldered one of my small TV terminals right onto the same board. It was a small computer with a small terminal, but it had good capabilities.

In our lab at Hewlett-Packard we had a desktop computer called the 9830 that ran BASIC. It was designed for the scientific community and cost \$10,000 so it wasn't a personal computer, but you could run BASIC as soon as you sat down. That was my goal with the Apple I—you could sit down, turn it on, and start typing.

That was the main thing about the Apple I. Its characteristics were largely centered on its video terminal capability. In those days, the most common input/output mechanism was the ASR-33 teletype. It had been a standard for 10 years, and the minicomputer companies had only recently started using video terminals. So I had lots of experience using teletypes,

and now I could do it in video with my terminal.

In 1975 video terminals were designed with shift registers because there were no cheap RAMs. You'd set up a bunch of shift registers and keep shifting them around to send text to the TV screen. So the Apple I was slow. It could type out 60 characters a second—one character per scan of the TV screen. My motivation was totally to save chips, not to add features.

BYTE: *Was the Apple I really a full-blown computer?*

Wozniak: Yes, but its features were a little bit different than the Apple II's and other personal computers that followed. It was slow, and it was text only, but it was a lot faster than the teletypes we were used to. They could only type 10 characters per second. TV terminals were just starting to get popular in those days.

BYTE: *Weren't they still fairly expensive?*

Wozniak: Yes, but I had to be cheap because I didn't have any money. I used the oldest, cheapest surplus parts I could find. Don Lancaster had written an article called "TVT1" for *Radio-Electronics*, and it was the big hobbyist article of the day. He had a humongous design that used tons and tons of parts and gates, but it was a poor design. I was into tight, clever designs to cut chips down, so I was flexible about the video timing. I knew from my high school days that TVs are designed with a lot of slop. Even if my timing was a little bit off, it would still work with most TVs and monitors.

BYTE: *So you didn't worry too much about precise timing?*

Wozniak: No, I was in a very hobbyist realm. I wasn't designing a product, just something that would work at home on my own TV. The computer itself used a 6502 processor interfaced to the terminal through a parallel interface chip called a PIA. It could also read a keyboard, so I bought a surplus keyboard that was advertised in an electronics magazine for \$60. It emulated an ASR-33 teletype and did everything I wanted it to do, so I interfaced it.

Memory was my main problem. The only surplus RAMs available at the time were 2102s—1K static RAMs. I had the computer designed and the BASIC written, so I borrowed a 4K 2102 static RAM board from a friend so I could test it out. I got my BASIC going with that, but I wanted to use dynamic memories because you could cut the chip count way down.

Steve got intrigued with all these ideas and one day he asked me, "Why don't you use these new 16-pin dynamic RAMs?" I had looked at them in my work at Hewlett-Packard, but they were new and I couldn't afford any parts that didn't come my way almost free. I'm a little bit shy, and I didn't know any of the reps, but Steve just called them up and talked them into giving us samples. I jumped on it. I thought it was a great part because you could replace 32 chips on a board with just eight. It was a little more difficult to use because you had to multiplex the row and column addresses and that cost me one or two chips. But I was very happy because it was TTL [transistor-transistor logic] compatible and I could save a lot of board space because the parts were so much smaller. My whole goal was to make it as small as I could. I now had a little computer on one board about six by eight inches that

*'Showing off Breakout on the
Apple II at the Homebrew
Computer Club was the most
satisfying day of my life.'*

I could take down to the club and show off. It was only 30 or 40 chips and it could run BASIC. People would look at it. It was unexpected.

BYTE: *So how did Apple Computer actually get started?*

Wozniak: Steve and I had both been going down to the [Homebrew Computer] club and giving out schematics for the computer and the terminal, even going over to people's houses and helping them build and test the computers out. Steve said, "Look, people are interested in what you've got. Why don't we make a PC board, have it silkscreened so they know what parts to plug in, and sell it at the club?"

We had about 500 members in the club, and I thought that maybe 50 people would buy it. It would cost us about \$1000 to have the board laid out, and each board would cost us about \$20. So if we sold them for \$40 and 50 people bought them, we'd get our \$1000 back. It seemed pretty doubtful. But Steve said, "Well, yes, but at least for once in our lives we'll have a company." So Steve sold his van and I sold my HP calculator to raise the money to make the PC boards.

Right away Steve got a big order from a local computer store to supply completely built computers. They ordered something like 100 units at \$500 each, to retail at \$666. It was unbelievable—a \$50,000 order. We were in business.

All of a sudden we needed about \$20,000 to buy parts. Steve went down to the local parts suppliers and we filled out financial forms. They looked at our purchase order, made phone calls, and really looked into our customer's credit. In the end, they gave us the parts with 30 days' credit. We had everything set up to build the computers and deliver them in 10 days, and it worked out great. We delivered the computers, paid off the parts suppliers, and only had to borrow \$5000 from a friend, Alan Baum, and his father.

BYTE: *How many Apple Is did you actually sell?*

Wozniak: We manufactured 200 of them and sold all but 25 over a period of nine or ten months.

BYTE: *When was that?*

Wozniak: In 1976. It was first demonstrated running BASIC in late 1975, and Steve suggested forming a company in late 1976. We formed an official partnership in March. We had a third partner at the time who had 10 percent, but he sold out for \$800 because he thought we weren't going to go anywhere except into debt, and he was the only partner with money.

BYTE: *Just to put four or five stories to rest, where did the name Apple*

Computer actually come from?

Wozniak: It came out of Steve Jobs's head, and he's a sort of private person, so I can't say what led up to it. He came up with an inspiration. He was working from time to time in the orchards up in Oregon. I thought that it might be because there were apples in the orchard or maybe just its fructarian nature. Maybe the word just happened to occur to him. In any case, we both tried to come up with better names, but neither one of us could think of anything better after Apple was mentioned.

BYTE: *Hewlett-Packard didn't want the rights to the Apple I. You designed it while you were working there. Did you offer it to them?*

Wozniak: Yes. There were a few of us in the lab at HP who were interested in microcomputers. We had proposed it to the lab manager. We sat down and had a meeting, and worked out on paper how we could have a little \$800 machine that could run BASIC and connect to a home TV. Now this guy had been the project manager of HP's 9830 desktop BASIC machine, and he had been through a lot of these issues. He knew why this couldn't be an HP product, and he was right. Hewlett-Packard just couldn't do a hobby product—they just couldn't get into an evolving market when it was too young and unforeseen. So he turned it down and I got a legal release. A funny thing happened. After we started shipping Apple Is, our calculator division at HP started a little 8-bit processor project called Capricorn. I had already done most of the stuff they were doing, but they wouldn't let me work on that project.

BYTE: *Can you summarize the characteristics of the Apple I?*

Wozniak: It used the 6502 and included 8K of RAM by the time we put it out as a product. You could load BASIC into 4K of memory and you had 4K left over for your own BASIC programs. We supplied the boards completely put together and it had a video connector, but you still had to connect a video monitor on your own. You also had to get a keyboard and wire it into a 16-pin DIP connector. We built a power supply onto the board, but you had to connect two transformers for 5 volts and 12 volts. It had no speaker, no graphics or color. It could just display text at 60 characters per second.

BYTE: *The display wasn't memory-mapped?*

Wozniak: No, it was just a computer with a built-in video terminal. You could hand-wire certain addresses to read the keyboard, write to the video terminal, and select addresses for 4K memory bank 1 and bank 2. It had one external card connector, and we quickly filled that with a cassette interface card. It also had a side connector where we could theoretically add something later to expand it. The idea was just very small and simple, and it was all on one board.

BYTE: *Was there a monitor in ROM?*

Wozniak: Not really. In those days PROMs were very expensive, so I just used two small 256 by 4 PROMs to give me 256 bytes of ROM. I wrote a little program that would let you type in hexadecimal codes, examine a range of memory, or run a program at a particular address. Those three functions fit in 256 bytes.

(continued)

THE APPLE II

BYTE: *How did you make the transition from the Apple I to the Apple II?*
 Wozniak: We were selling Apple Is and we were just having fun and getting known. It was the most incredible thing we had ever done in our lives. I was still working at Hewlett-Packard and just moonlighting at Apple to test out boards, write more programs, and design a cassette interface so you could load BASIC in just a few minutes. [*Before the cassette interface, you had to type in the entire 4K BASIC interpreter by hand, in hexadecimal. . . . R.M.*] I had taken the machine down to the Homebrew Computer Club for its official introduction and pointed out its features. I got asked a lot of questions—people wanted to know if it could do other things.

At the time, Cromemco had just come out with a board called the Color Dazzler that did color graphics on S-100 systems. We had also had a demo, at the club, of a minicomputer running a display of a clock on the screen in color. During one of my times at Atari with Steve, I had designed a little seven-chip circuit to do color on a TV screen and it worked. So I started working on things that I wanted to add to the Apple I. I was thinking about clever color circuits and how to cut the chips down.

Remember, I had a computer with its own memory and a terminal with the display memory in shift registers. So I started thinking how I could combine parts into one—to somehow get them both to use the same memory. I finally worked out a design where a small part of the main memory was used as the terminal's video memory.

Eventually I got the whole machine done. It did everything that the Apple I did, except the display was in main memory so you could change any location on the screen instantly. I had built-in color, and EAROMs [electrically alterable read-only memories] were getting more common, so it had software built in to make it operate like a terminal. So it had color, it was very fast, and it was still cheap. In the end it was about half as many chips as the Apple I and it was many times better.

BYTE: *Was the designed-in color low resolution or high resolution?*

Wozniak: At first it was low-resolution color. Basically it could output characters in two different ways, either as text or as colors. I actually had to modify our horizontal video rates to be a little different from the NTSC [National Television Standards Committee] standards, because NTSC was not designed for digital video.

Once the computer was done, I started writing routines for the ROMs. I wrote terminal software so the characters would display in rows, from left to right and move down to the next line for a carriage return. I expanded my monitor routines to do more than just load hex, display memory, and run programs. I added the cassette routines because I knew they were going to be important. Eventually I started adding fancy features like a disassembler and the ability to split the screen into different windows. I would just go down to the club every two weeks and show off the latest routines I had done, because it was impressive. It wound up being a 2K monitor with graphics routines and everything included.

All my thinking from the Apple I days was carried over into the Apple II. Most people could not afford a color monitor, so it had to work with a home TV. A home TV could only

display 40 characters across the screen, and I had to do what a TV could display. It was interesting because the technology defined the product rather than the market defining the product.

BYTE: *What caused the odd mapping of the graphics addresses?*

Wozniak: I had horizontal and vertical counters in the video circuits, and I had to map them into the display memory addresses. If you think about it, it's simple to have a perfectly linear address space. You take the vertical count times 40 and add it to the horizontal count. But it took three 7483 adder chips to do that, and I was looking for ways to save chips in the design. I figured out a couple of tricks that let me do it with just one adder, so I was able to save two chips; that helped others at the Homebrew Computer Club.

Also, I assumed that the user would just send characters to the screen through the terminal routines in the monitor. He would never see the odd addressing. I just wanted to play games and run BASIC. I didn't perceive that anyone would want to address the screen directly. Fortunately, only the vertical addresses were out of order, and there are a lot of easy ways to handle that so it doesn't slow things down.

BYTE: *How did you come up with the scheme where each byte only displays 7 bits, and when did the high-resolution graphics arrive?*

Wozniak: I had used the cheapest parts available, and a 5 by 7 character generator was the only cheap one in 1975. So my characters were five dots across and they fit into a 7 by 8 matrix. From my experiments with color and my experience with TVs, I knew how much time was available to display data during a TV scan. I also knew that my master clock had to be a multiple of the color-burst frequency, so I used 14.31818 MHz, four times the U.S. color subcarrier. So with that clock, I just counted how many dots per character would get 40 characters on the screen. I could have used eight dots per character and displayed 32 characters per line, but if I tried to display 40 characters of eight dots each it would have been too wide to fit on the TV screen.

One day I mentioned to Steve that I had noticed something interesting in the video addressing. I could make a little change by adding two chips, and then I could just shift each byte out onto the screen and we'd have hi-res graphics. I wasn't sure that it was worth the two chips because I was very chip-conscious. But Steve was pushing for all the features we could get, so eventually we put it in. At the time we had no idea that people were going to be able to write games with animation and little characters bouncing all around the screen. It was a neat feature, so we put it in there. Well, now I had a computer that had color in roughly the size of the bricks in the Breakout game I had designed for Atari. I had written a bunch of assembly-language routines to draw spirals and zig-zags of color, and I would take it into Hewlett-Packard to show the engineers. Sometimes they would sit down and say, "This is the most incredible product I've ever seen in my life." Many people were saying that when they saw the colors, but it was not in BASIC yet.

I wanted to write Breakout. I knew I could do it in assembly language, but I hadn't put any graphics commands in the BASIC yet. I knew how to add commands, since I had written the BASIC, so I put in a command to plot simple color

INTERVIEW: STEVE WOZNIAK

squares and soon got to a point where I could draw a brick wall. Then I did a ball and wrote some routines to make it bounce around. Now I needed a game paddle.

I wasn't sure how to sense the value of a paddle. I didn't want to put in 20 chips or anything expensive. So I found some little timer chips—NE555 timers. They were very cheap, but they could sense the value of a pot [potentiometer] by controlling their timeouts. There was also a larger one that was four 555 timers on one chip. So it was trivial. With a few resistors and capacitors I could read four paddles. As I got further into the game I needed sound, so I put a speaker in also.

Basically, all the game features were put in just so I could show off the game I was familiar with—Breakout—at the Homebrew Computer Club. It was the most satisfying day of my life when I went down there. I got a couple of high school kids to help me set it up, and I demonstrated Breakout—totally written in BASIC. It seemed like a huge step to me. After designing hardware arcade games, I knew that being able to program them in BASIC was going to change the world. All of that stuff is the essence of the Apple II.

BYTE: *We've heard that someone else helped you with the design of the I/O slots on the Apple II. Is that true?*

Wozniak: Yes. Many of the computers of the day had I/O card connectors, but they all required a bunch of address-decoding circuitry on each card. It was costly and required extra chips. I was into low-chip-count designs, and I wanted to have eight slots. So I was thinking of dividing part of the address space up between the eight slots and just using one decoder on my board to decode the eight different addresses. In that way, each I/O board didn't need three chips' worth of decoders on it.

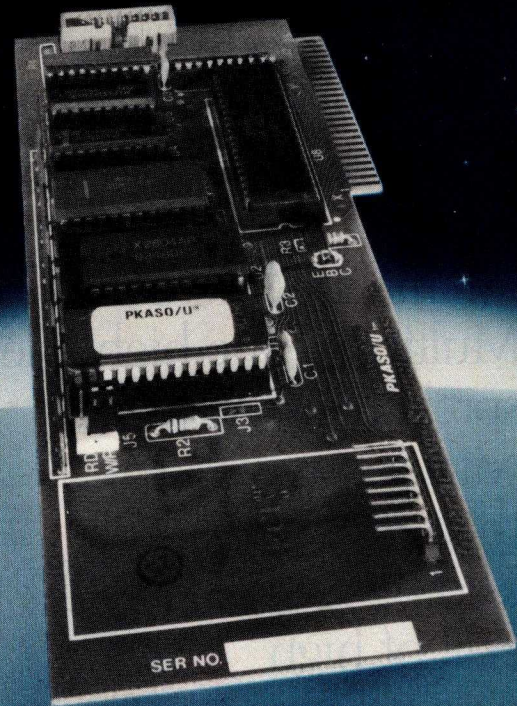
Alan Baum helped a lot. I already had some address decoders on the board that decoded every 16th and every 256th I/O address. Alan realized very clearly that each card could have its own little 256-byte PROM on it and how they could all share a bank-switched 2K space. So I could just send two decoding signals to each card. Each board had 16 addresses for its I/O and 256 bytes of space for its PROM. Alan was the most constructive person in terms of realizing very clearly just how well the hardware-software interaction would work.

I saw one of Steve Ciarcia's articles in a recent issue of BYTE. He implemented a new voice chip design on an Apple II, and he said it was "because you don't have to decode it on the board." That was exactly the purpose behind the I/O decoding Alan and I designed. Oddly enough, he and his father were the ones who lent us the \$5000 we needed in the Apple I days.

BYTE: *How did you raise the money to build the Apple II?*

Wozniak: By the time we did the Apple II, we had to build 1000 boards that cost \$250 each. We needed \$250,000 and we had nothing. Mike Markkula helped us write a business plan, and then he started perceiving that maybe this home computer market was really going to happen. He thought that home computers would hook up to your oven and refrigerator. Obviously this didn't happen, but he decided to join us as a third partner. ■

PKASO/U®



The Universal Printer Interface

- Universal for all standard parallel printers.
- Famous for Graphics (LoRes, HiRes, SuperRes) (Logo Compatible)
- Terrific for text (even rotates spreadsheets to print sideways)

One set of commands for all printers. One command changes character sizes. Create your own printing fonts, alphabets and symbols . . . bold face, underline, italics, subscript and superscript, HiRes Zooming.

PLUS:

FREE Utility and Demonstration Software Disk. CLEAR, comprehensive user documentation. PKASO/U . . . for all the reasons you need an interface.



Contact us for a list of Authorized Dealers near you.



Interactive Structures, Inc.
146 Montgomery Avenue
Bala Cynwyd, PA 19004
Telephone: (215) 667-1713