

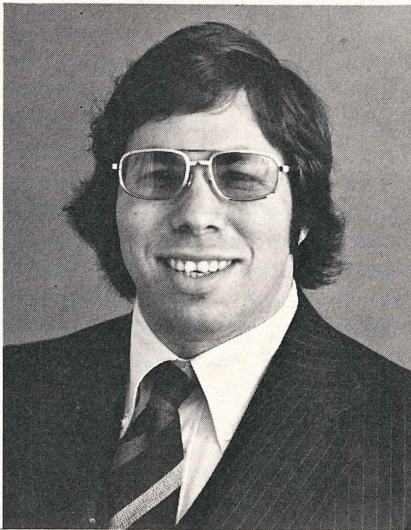
# CONTACT



the user group newsletter

## HOMEBREW TO CHAMPAGNE

by Steve Wozniak  
Apple Computer, Inc.



**Steve Wozniak**, Co-founder Apple Computer, Inc.

*In a recent speech before the International Apple Core, Steve Wozniak, co-founder of Apple Computer, described how his involvement*

*with a local computer club helped give birth to Apple.*

I'd like to compliment the organizers from the International Apple Core for bringing together so many Apple fans. I hope you all love your Apple II's as much as I love mine.

Those of you who read the magazines are aware that the Homebrew Computer Club was formed back in spring of '75. It was one of the first computer clubs in the country. The phases it went through, from formation to maturity, parallel what I was doing at that time, which also led to the birth of the Apple I and Apple II. Both products to a large degree were born out of the club, rather than from an existing company or product philosophy.

At the time, I was working at Hewlett-Packard designing calculator-type circuits. I knew how to design and write compilers and such, but I'd kind of let that slip for a few years. I was more interested in cal-

culator chips and driving around the world and just doing all the things you do at that age.

Well, I got involved in designing video games. I was in a bowling alley and saw a PONG game and thought, "Hey, I can design one of those." So I went out and designed my own version. It was famous around H-P because when you missed a shot, a random message would flash on the screen and say something like "Oh Shit!" or "Darn it".

I also got interested in video terminals, because I watched some friends using Teletype 33s, which were popular low-cost terminals in those days. I could talk to timesharing systems and play Lunar Lander and look at all sorts of interesting files.

I wanted a terminal, but couldn't afford one. At Hewlett-Packard they had a flexible policy which said, basically, that any engineer is welcome to parts for projects of his own design. So I took advantage of

that policy and designed a video terminal and a modem to get on various networks.

This was about the time the Homebrew Computer Club was formed. I heard from a friend that they were interested in microprocessors. I didn't really know what microprocessors were, but I decided to go, since they were also interested in video terminals.

We got together in Gordon French's garage. It was raining outside, but he had his garage door open and 40 of us were crowded inside. We introduced ourselves around the room. A good many of those people later started their own personal computer companies—companies like Processor Technology, Cromemco, and some of the other peripheral designing houses.

I learned a lot that night. I was a little bit behind these goings on. There was some discussion about whether we should start an 8008 portion of the club because Gordon French had one. He had a bunch of software he was writing for it and he passed out 8008 instruction manuals at the meeting. It's probably a good thing we never went in that direction, but focused more on the 8080, which was pretty expensive back in those days, about \$270.

That was the formation of the club. Exciting things were happening and we would continue to meet.

At about that time, I had a video terminal and some video games. My terminal was almost complete. Lee Felsenstein, who later became our chairperson, also had a video terminal—it later turned into a Processor Technology board.

I was amazed by the high school kids at our meetings. Everyone knew the buzzwords, knew what was going on and which chips had what kind of instructions, saw all this home/hobby computer business through the Mark 8 to the ALTAIR. I didn't know anything. I had kind of drifted out of my early high school and college experiences in minicomputers, and I felt like a dummy. All these other people knew what was going on. I was certainly interested in what they were interested in. It sparked my interest in small computers, minicomputer-type instruction sets, and microprocessors.

Forming a newsletter is critical to the survival of a club, because that's how you get the word out and keep the interest up from month to month. The first couple newsletters of the Homebrew Computer Club were typed on one sheet of paper, just a list of names and addresses. We also set up a library. Gordon French offered to maintain a repository of software and articles and whatever else. He also offered his phone number between certain hours on certain nights. We could call him up and say: "What's going on? Do you have one of these?" It was really a large task for one person, so it kept him busy for much of the next year.

One of the early topics of discussion, while the club was being formed, was how home computers were about to happen. Everyone in the hobby clubs was saying it, but no one anywhere else was. At work, I would walk around and talk to guys in the lab, and everyone there was the way I had been, isolated from what was about to happen. Possibly some of the concepts of the hobbyists in those days were a little off base from what really happened, but it was obvious even then that the hobbyists were going in the right direction.

Byte magazine was the first to hit the stands. I was fortunate enough to have been one of the first people to buy it. People at Homebrew were talking about Altair Computers—who had one, who had ordered one, and when everyone was going to get one. Very few people in the club at that time had their own computers or could afford to build one.

The only way to have a computer was to build your own, which a few people had done. I was putting some of the finishing touches on my video terminal, and starting a file of all sorts of microprocessor-related information. There were only two microprocessors at that time, the 6800 and the 8080. Since I was familiar with minicomputers, the 6800 struck me as a more intelligent machine to go with.

I started collecting thousands of data sheets on terminals, printers, breadboarding accessories, and all the things that fit into the microprocessor world. Then the Homebrew Club went into a phase of very rapid growth. There's probably a few of

you in this room who were there and remember it. It was extremely exciting. Each two-week meeting was the most important thing in your life. Rumors were flying. People were throwing ideas up in the air and guessing what was going to happen next.

Interface Age magazine was introduced about this time. It had started as a southern California computer club magazine. Magazines like Byte and Interface Age really helped make the whole thing happen on a national level, because everyone was getting informed. Without those magazines, it might have taken years for this thing to have become a full-fledged major industry.

A few other clubs were being formed around the country, and the magazines would list them. Now there are far more Apple Clubs alone than the total number of clubs in those days.

Very few companies were making hobby computers back then. MITS and SPHERE and a couple of others. These companies were all small enough so that they could afford to spend time with us. And since some of them were local, we were able to see some of their very first demos even before the equipment was in the stores. We would see some of the early equipment, like the first IMSAI.

A lot of the club members were interesting personalities. We had fellows who would ask around the room, "Is there anyone here from Intel?" If there wasn't, they would donate a bunch of Intel microprocessors to a raffle. We had our Rona Barrett types, too, who managed to keep the rumor mills spinning. I would say the rumors were 90 percent inaccurate and 10 percent accurate, but it was always interesting because they were digging up information.

Meetings were like a free forum; anyone could stand up and say what was going on. We had a few used chip salesmen, who were hocking chips at low prices and various other pieces of equipment like motherboards and extender parts. We'd have people who would get up and say, "I'd like to help anyone with hardware or software." And they usually got latched onto by someone who really needed a program written. There were people who would say, "Hey,

I need help in this area, can anyone help me or locate a part for me?"

If you think about it, the purpose of the club, more than anything else, was to facilitate communication. Lee Felsenstein, our chairman, organized the club so that we had what he called a "mapping period." Those who wanted something and those who had would stand up and say what their need or offering was. They would sort of "map" together and spot who they needed to meet with during our "random access period," which was a get-around to talk with people who had the same interests.

The Club grew quite quickly until we had about 550 members meeting twice a week. This was an extremely large number of people meeting for something that wasn't even making news.

People also set up demos of equipment they were working on. That was an opportunity for me to show off my video terminal. During this period I was designing a 6800 system, not having any hardware but knowing Hewlett-Packard had decided on the 6800 as their "standard" processor.

At the time, I could get some at a reasonably low cost—like 40 bucks per chip. There was no way to get an 8080 that cheap in those days. So I designed a system around the 6800, based on timing diagrams. When you're a computer designer, you have to look at timing diagrams and become intimately familiar with how many microseconds it takes to get from this phase to that event and such.

About the same time, which was late summer of 1975, a company called MOS Technology began making news by claiming they had a better processor than the 6800, that had been developed by the same people that designed the 6800, was hardware compatible with it, and was going to be sold over the counter (which was absurd) for \$20. It was amazing, because in those days you always had to deal with major companies. Go through reps, stocking distributors, and the like. And here MOS was, selling microprocessors over the counter. But a lot of people were getting interested in this kind of thing.

I and a couple of other Hewlett-Packard people who were inter-

ested in getting microprocessor chips took this opportunity to visit the MOS Technology booth at a trade show and buy their new microprocessor—the 6502.

I can still remember the bus ride home. We were looking through the manual and discovering what the 6502 did and didn't have. That night, I went to a meeting of the computer club. The people from SPHERE came by and told us they were about to introduce the Micro-SPHERE, which they never really did. They also demonstrated a larger computer which was like a Nova. It had a hard disk with some color graphics running to a color TV display. In those days, on a hobby level, this was unheard of. I was very impressed by a color clock on the screen, and all the different colors, so I got interested in color graphics. At the time, it was more like, "Hey, this is really neat. But it's impossible to do because I know about TV, and it's difficult to just go out and design a color circuit."

Once I had the 6502, I started thinking about what I could do with it. I could have gone ahead and built a machine, but I was more interested in going through the manual and writing some code in the native machine language of the processor. I was used to assemblers but, unfortunately, there were no assemblers available. So I learned a technique that I didn't even think was possible: I just wrote the code down on paper, filling in the op codes and the displacements.

The first thing I wanted to write was a BASIC interpreter because it was a hot, catchy language and I wanted to do the first one for the 6502. It seemed like an interesting, worthwhile project, but it wasn't like it was going to be a product or something, just something to show off to the club.

I had never programmed in BASIC, although I knew it was so close to FORTRAN, which I could use, that I'd have no problem with it. BASIC seemed to be the favored language of the club people.

I'd have loved to write a compiler, but it was more important to get it written than to have a full language. I made tradeoffs to keep the code very small and went with an integer language. Because I had no assembler, I was also forced to have a loosely bound program with all the

modules intercommunicating. With that kind of code, you need make only one change and then, everything has to be reassembled. I just could not take the time to reassemble the code myself by hand. That forced me into a structure where every new feature in the language was added on as a module, which actually made the development go very quickly and smoothly.

I got the interpreter developed and running on some simulators at H-P. Then I thought, "Hmmm—I don't have any hardware." As far as 6502s were concerned, nobody had anything in their hands except a chip. So I pulled out my own 6800 design, made the necessary changes to accommodate the 6502, and came up with a reasonable system that would get me to a point where I could connect the keyboard and the video display.

There were few processors in those days that had keyboard entry. This is what people were beginning to do with their ALTAIRS. I had already been through the front-panel switch stage and decided from day one that you really want a keyboard and video display.

The computer outfits back then did not have such things as monitor ROMs. It's a crucial thing to have, at least for an Apple II type machine. So I wrote a monitor. I kept my design small, so it only took me a couple of nights to get it wired together by hand and tested. I used my existing video terminal, which was a slow speed terminal but compact enough to stick on the same PC card as the 6502 processor.

So I got the computer together, and borrowed a four-case static card from a friend who had wired his own card together out of 2102s. And I managed to get it to play and got my compiler up, so I was able to sit down at the keyboard and test things out with a miniscule monitor. The circuit eventually became the Apple I.

What helped speed up the development was that every two weeks I could go back to the club and talk to people and say: "This is what I'm doing," and they would listen with interest.

In November, 1975, I took it down to the Homebrew Computer Club and ran BASIC for the first time. Of course, there were no cassette or even teletype loaders, so I

had to type in the text by hand. After about an hour, I had enough of BASIC working that I could run some interesting programs. Some high school kids who later became part of the early Apple crew would sit down and type out demo programs. A lot of people were interested because the computer was small and compact.

Shortly after this, I decided this 4K RAM board was not the way to go because everyone was starting to introduce 4K dynamic RAM chips. They were much more dense and compact than the 4K RAM board. So I got some AMI, three-transistor, cell design parts, surplus at a good price. The part was doomed not to be made very long. It was a 22-pin package, which meant you could connect all your address lines to the RAM, but you had to design special transistor circuits. You know, I'm a digital person. I like digital chips, but on these you had to design special clock drivers.

I brought the card up on those dynamic RAMs and it was very impressive to take it down to the club and show people I was using the new 4K chips. In the hobby world, you just didn't do anything like that when they were brand new.

Steve Jobs got me some and said to me "Why don't you use the 16-pin chips?" I looked at all the data sheets and found it was easy to interface because you didn't have to design an expensive clock driver and other types of logic to support the dynamic RAM.

I put these dynamic RAMS on the same card with the video terminal and processor. It was really quite a thrill to look at this entire board and know it was a usable computer. A lot of people were still talking about waiting to get their ALTAIRS and switches and maybe buy some peripherals someday.

About this time, those of us who were interested in the home computer revolution approached management in our lab at H-P and said: "Hey, look—with so much RAM, so much microprocessor, so much video, and a case, you can build a computer that talks BASIC and sell it for a thousand dollars." H-P had a lot of reasons why it couldn't be an H-P product. The lab manager said it was a great idea for a start-up company but wouldn't work for H-P. Later, I went through the legal

department to get a release for the Apple I, which meant that no H-P division was interested in it.

Meanwhile, back at the Homebrew Computer Club, we had moved into a big auditorium. My favorite place was in the back row so I could set up my Apple I stuff while the show was going on. Everybody there was into this idea of the computer being a motherboard, a processor card, and a bunch of 4K RAM cards, and maybe a video card and an interface card. It was just not integrated enough and they looked just like all the minicomputers, with lights and HEX and switches on the front.

It was a big advantage to have a small, low-cost, integrated computer on one PC board, because people got a machine where they could talk to the screen and really do useful things in BASIC.

I started passing out schematics of the thing, since we didn't have a product or a company at the time. I'd go to people's houses and help them wire up their own. Steve Jobs said, "Why don't we just form a company and sell PC boards? You know, people really want these things, because hand writing is a major operation and a PC board is very inexpensive." A lot of people were offering PC board level computers in those days, so I decided we'd do it.

I figured we wouldn't make any money at it, but I sold my H-P65 calculator (partly because I knew they were coming out with the H-P67) and Steve sold his van, and that gave us our starting capital to pay a PC designer to design it. All the user had to do, once he got the board assembled, was plug in transformers, a video monitor, and a keyboard. In those days, it was unheard of for a hobby-type product to be that completely assembled.

Steve managed to sell some completely loaded boards, and all of a sudden we were in the computer business. The ALTAIR bus was still the big thing in the hobby clubs: People would look at Apple and say, "They're just a garage shop."

If this thing had been designed originally as a product, a lot of different decisions might have been made, such as to use the 8080 or go with the S100 bus, and a certain type of product that now makes

sense in retrospect wouldn't have happened.

By the time we made our first delivery, we were in desperate need of a cassette interface. We ended up designing our own, and that pretty much gave us the Apple I system. If anyone has one, I wouldn't trade it in on an Apple II because it's a rare item now. We only built about 200 of them, and we only shipped about 175.

The Apple II came out shortly after. I had been thinking about how to put color on the Apple I and about carrying the designs and the concept a bit further. By the time I had color graphics up and running, I would take it down to the club every two weeks and show it to people.

We managed to get hold of some of the first 16K RAMs coming out, because of Steve Jobs's connections, and we shipped a couple of 32K Apple I systems using 16K RAMS. I believe they were the first computers ever shipped with 16K RAMS, because the manufacturers couldn't supply any of the big companies yet.

A lot of Apple I and Apple II enthusiasm was spurred by the positive feedback I'd gotten at the club. It was an interesting atmosphere in which to develop a product.

The main function of the clubs is to facilitate communication on various levels, and the International Apple Core appears to be a very good structure to disseminate information to everyone.

#### **Question:**

Why didn't you go with the S100 bus on the Apple?

#### **Answer:**

If the decision had been made: "Let's start a company and sell some products into this market," we would have gone with the S100 bus. What happened was that I was trying to build a small computer for myself. And when you keep it small and on one PC board, it just led to a more natural: "The microprocessor bus has enough information, why go with the 'standard' bus?"

It would also have meant going with an 8080 processor, which I didn't have and couldn't afford. If there had really been an S100 bus on the Apple computer it probably wouldn't have destroyed the Apple

concept. It just wasn't necessary, and there are enhancements to our buss, such as pre-decoded I/O, that turned out to be much more favorable. So I'm kind of glad we didn't, considering that of all the \$100 companies around, just a few emerged as solid companies.

**Question:**

One of the things that makes a company really great is allowing someone with an idea, regardless of how off the wall it is, to play with it. Does that kind of thing exist at Apple or is the pressure reaching a point where you're on a project and that's it?

**Answer:**

That kind of development is very much encouraged. A lot of people at Apple like myself, Bob Bishop and some of the others, have just come up with off the wall concepts overnight or a weekend idea, a weekend project, and there's absolutely no problem unless it gets out of hand. I would encourage it as long as the normal work gets done. We have several specific examples where that does happen. A lot of our engineers come out of a club environment where they did things on their own anyway to begin with, which was also good.

**Question:**

Remember, that in the design days memory was really expensive and more characters meant more money. A 4K machine was a big machine and we couldn't afford to supply a video terminal with each Apple. We wanted to cut that cost down so we decided, hey, we'll just put video out and modulate it and go into a TV set and it's a lot less expensive besides, everyone has a home TV. You can't put more than 40 characters reliably on a TV, it's that simple, the band width of the RF units won't cut it. That's how the decision was made. If we knew we were going to do a system that was going to be used for small business, word processing and whatnot, the decision might have gone a different way.

**Question:**

Is it time to think about the fact that everyone's getting a VCR?

**Answer:**

It's funny, but the VCR market isn't growing as rapidly as anticipated, although it seems around here that it is. But you read the marketing reports and they're not selling many. Maybe in other parts of the country. Almost everyone I know has a VCR. I think that the VCR is a great mechanism. I think that there's room for some people to design some very simple circuits that can transfer data at very high rates with sufficient redundancy to store large data bases on a VCR and even randomly access it. VCRs all have the circuitry so you can rewind it, etc., so you might think of a VCR operating system. It's possible. It's feasible. One of the problems you get into is that, whenever you have that kind of operating system, you have to have gaps between data fields. Let's say you have a VCR you can start and stop. Well, the stopping has to be short relative to the data time or your tapes going to only be ten percent full of data. It turns out that in a few inches on video tape you can store far more than the amount of memory that your system contains. There are some problems but I wouldn't be surprised if it happens anyway. I don't know if you're aware of what CORVUS has done with their MIRROR system, but they back up their hard disk on video tape in 10 minutes. That's a \$600 backup scheme so you don't have to use 100 floppies or buy another CORVUS drive to backup the one you have.

For

Subscription

Information

see page 95



RESIDENT FIRMWARE UTILITY PROGRAMS WILL EXPAND THE CAPABILITIES AND INCREASE THE USEFULNESS OF YOUR APPLE II OR APPLE II PLUS! \*

**\*APPLESOFT EDITROM**

Global search, change, or remove any string, variable, literal, constant, or basic command word that appears in your Applesoft Programs. **EDITROM** uses no ram space that will interfere with your program. It does not reset any system pointers to protect itself and will operate with any size system - 16K, 32K, or 48K. After **EDITROM** has been initialized, the ampersand (&) command can then be used to call the **EDITROM** back for repeated use without readdressing the ROM+ BOARD. Completely compatible with Konzen's Program Line Editor. If PLE is up, **EDITROM** will keep PLE up and allow joint operation. Will operate with any version of **DOS** and requires FP in Rom. 35.95

**COMMANDROM**

**COMMANDROM** is like having a resident 'FID' but with more operating features and conveniences. **COMMANDROM** will read a disks (13 or 16 Sector) File Directory Listing and display the following: A Command Menu, current drive number, number of sectors used and left, number of pages set up to hold all file names, the first page of file names, and an identification letter next to each file name on display. Pressing any one of the command keys will load or run any file (A'B' file load will display start and length addresses.), lock or unlock a file or all files, delete a file - with verify before deletion, change from one drive to the other, read a new disk, display a Track/Sector Map, change page numbers to view all file listings without recataloging, or exit to current language or monitor. No system pointers are reset and no RAM is reserved for **COMMANDROM**. Requires 48K, 3,2 or 3.3 **DOS** and **INT** or **FP** in **ROM** 35.95

**BASICSROM**

Will boot a 13 sector disk on a system configured for 16 sector operation. The **BASICSROM** can be addressed on coldstart (without Auto Start **ROM** or warmstart (with Auto Start **ROM**) at any time 35.95

**DISK COPY/SPACEROM**

Duplicates a disk, from either **SINGLE** or **DUAL** drive, single or dual controller, 13 or 16 sector and with or without **VTOC**. Options include - Gross copy, active sectors only copy, **DOS** overwrite, auto boot of copy disk, free space on disk in sectors and kilobytes, and Init and volume number are selectable. Requires a minimum of 32K 35.95

**\*APPLESOFT RENUMBER/MERGEROM**

Made famous by Apple Computer Inc., this powerful utility will not disturb any part of a program in memory, when it is activated. Requires 48K, with or without Disk II. 35.95

**'YOUR' PLE ROM**

Now you can put your Program Line Editor in **ROM**. Write for details. \$45.95 \*REGISTERED TRADEMARKS

BOX 599  
WEST MILFORD, N.J. 07480  
**SOFT CTRL SYSTEMS**