# FOUNDERS AT WORK

STORIES OF STARTUPS' EARLY DAYS

*Jessica Livingston*

Apress®

*Founders at Work: Stories of Startups' Early Days*

Copyright © 2007 by Jessica Livingston

Lead Editor: Jim Sumser
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Matt Wade
Project Manager: Elizabeth Seymour
Copy Edit Manager: Nicole Flores
Copy Editor: Damon Larson
Assistant Production Director: Kari Brooks-Copony
Compositor: Dina Quan
Proofreader: Linda Seifert
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Library of Congress Cataloging-in-Publication Data

Livingston, Jessica.
Founders at work : stories of startups' early days / Jessica
Livingston.
p. cm.
ISBN 1-59059-714-1
1. New business enterprises--United States--Case studies. 2.
Electronic industries--United States--Case studies. I. Title.

HD62.5.L59 2007
658.1'1--dc22
2006101542

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

CHAPTER 3

# Steve Wozniak

## Cofounder, Apple Computer

*If any one person can be said to have set off the personal computer revolution, it might be Steve Wozniak. He designed the machine that crystallized what a desktop computer was: the Apple II.*

*Wozniak and Steve Jobs founded Apple Computer in 1976. Between Wozniak's technical ability and Jobs's mesmerizing energy, they were a powerful team. Woz first showed off his home-built computer, the Apple I, at Silicon Valley's Homebrew Computer Club in 1976. After Jobs landed a contract with the Byte Shop, a local computer store, for 100 preassembled machines, Apple was launched on a rapid ascent.*

*Woz soon followed with the machine that made the company: the Apple II. He single-handedly designed all its hardware and software—an extraordinary feat even for the time. And what's more, he did it all while working at his day job at Hewlett-Packard. The Apple II was presented to the public at the first West Coast Computer Faire in 1977.*

*Apple Computer went public in 1980 in the largest IPO since Ford in 1956, creating more instant millionaires than any other company up to that point.*

*The Apple II was the machine that brought computers onto the desks of ordinary people. The reason it did was that it was so miraculously well designed. But when you meet Woz in person, you realize another equally miraculous aspect of his character. A programmer might describe it by saying he's good in hardware.*

**Livingston:** Take me back to before you started Apple.

**Wozniak:** Even back in high school I knew I could design computers with half as many chips as the companies were selling them with. I taught myself, but I had taught myself in a way that forced me to learn all sorts of trickiness. Because you try to make valuable what you're good at. I was good at making things with very few parts by using all sorts of tricks—almost the equivalent of mathematics—so I valued products that were made with very few parts.

That helped in two ways. When you are a startup or an individual on your own, you don't have very much money, so the fewer parts you have to buy, the better. When you design with very few parts, everything is so clean and orderly you can understand it more deeply in your head, and that causes you to have fewer bugs. You live and sleep with every little detail of the product.

In the few years before Apple, I was working at Hewlett-Packard designing scientific calculators. That was a real great opportunity to be working with the hot product of the day. But what I did that led to starting a company was on the side. When I came home from work, I kept doing electronics anyway. I didn't do the same calculators we were doing at work, but I got involved through other people with the earliest home pinball games, hotel movies . . . The first VCRs made for people were actually made by an American company—not Betamax, it was before Betamax even—called Cartravision. It was put in some Sears TVs. I got involved with that. I saw arcade games—the first arcade game, Pong, that really made it big—so I designed one of those on my own. Then Atari wanted to take my design and make it the first home Pong game. They said to do one chip, which was better for the volumes that they would have—to do a custom chip. Steve Mayer came up with that idea. But I was kind of in with Atari and they recognized me for my design talents, so they wanted to hire me.

**Livingston:** How did they know you?

**Wozniak:** Steve Jobs worked there part-time. He would finish up games that they designed in Grass Valley. He brought me in and showed me around, and Nolan Bushnell offered me a job on the spot. I said, "No, I'm never going to leave Hewlett-Packard. It's my job for life. It's the best company because it's so good to engineers." It really treated us like we were a community and family, and everyone cared about everyone else. Engineers—bottom-of-the-org-chart people—could come up with the ideas that would be the next hot products for the company. Everything was open to thought, discussion, and innovation. So I would never leave Hewlett-Packard. I was going to be an engineer for life there.

Then I designed a game for Atari called Breakout, and that was a really incredible product. That was just so neat, to have my name associated with a product that actually came out in the field in video games. Because this was the start of a whole industry and I wasn't really a part of it. But I wanted to be a designer and just have some little connection to it.

In doing all those projects, I got involved in another one. The ARPANET then had about a dozen computers connected with a network. You could select which computer to visit, and they had certain access that you could get into as a guest; or, if you had passwords, you could get deeper. I just saw somebody typing away on the teletype, just talking about playing chess with a computer in Boston, and I said, "I have to do this. I just have to have this for myself." For a lot of entrepreneurs, they see something and they say, "I have to have this," and that will start them building their own.

I couldn't really afford to buy the pieces I needed. I couldn't buy a teletype, so I had to design my own terminal. The only thing that was free (because I had no money) was a home TV to see characters on. I got a keyboard for $60, which was amazingly low-priced then. That was the most expensive thing to getting my terminal built. Then it was just a matter of designing logic to put dots on a TV screen that add up to the letters of the alphabet and spell out what's coming from another computer far away. The keyboard types the data to the computer far away, and I built a modem for that. So now I had a TV terminal. This is while I'm working at Hewlett-Packard. I'm just doing these things on the side for fun in my apartment in Cupertino.

Back in college, I had designed a neat deal called a blue box, for making free phone calls. Steve Jobs came along and said, "Let's sell it." So now I had this video terminal, and he said, "There's a local time-sharing outfit that buys these expensive terminals. Why don't we sell this to them?" So we actually sold some of the video terminals that I had built. It was to become a portion of the Apple I.

I had wanted a computer my whole life. Back in high school I told my dad, "I'm going to have a computer someday." And he said that it cost as much as a house—the down payment on a house. And I said, "Well, I'll live in an apartment." But I was going to have a computer someday. So it starts with a huge dedication. You start with a lot of motives and values and who you are going to be in life. You start with those very early—some of mine even go back to elementary school. I decided there that I was going to be a fifth grade teacher, and I stuck to it and was. But some of these things you want so badly in life that, when the door opens, you are going to get there.

Now, I still was in this mode where I had to build everything for free. Then I discovered that microprocessors had come out. I had sort of slipped out of the electronics world, out of the computer world, due to working in calculators at Hewlett-Packard. All of a sudden I discovered these microprocessors. What are they? I didn't quite understand it fully, so I took a datasheet home.

There was a club that got started up. It was a club of young people—every one of them could have been an entrepreneur—the sort of people that liked to put together gadgets at home and make them work. But it turned out that not very many of them were real engineering designers that actually sat down and designed new things. Maybe they had jobs as technicians at work wiring stuff up, analyzing it, spotting inputs that were the wrong voltage. They were that kind of electronics person, but most of them weren't designers.

**Livingston:** This is Homebrew right?

**Wozniak:** This is the Homebrew Computer Club. There were a lot of software people that had no hardware background, and it took hardware to build these first machines. I was embarrassed because the world had somehow jumped ahead of me—they had come out with little cheap microcomputers based around microprocessors and I hadn't heard of it and I hadn't been a part of it. I felt very weird—that was the direction in life that I was going to be a part of when it happened. Well, I analyzed what a microprocessor was in one night, and discovered it was just like the minicomputers I used to design back in high school that were so good.

Then I looked at the Altair computer that started the whole thing going. It was the first microcomputer, but it wasn't really a computer. To me, I needed one thing. In high school, I told my dad that I was going to have a 4K Data General Nova. Why 4K? 4K bytes of memory. The reason is that's the minimum computer to run a programming language. You've got to be able to program in Fortran or Basic, or some language to get your programs done. The Altair that was being sold at a ridiculously low price, all it was was a glorified microprocessor from Intel, with some chips to protect the voltages. All they did was bring it out and say, "You can now plug in all the things that a microprocessor is designed to have added to it." You can add RAM, you can add cards that know how to talk to teletypes, you can add a big cable over to a teletype, you can buy a teletype for thousands of dollars. By the time you added enough RAM and everything else to have a computer that would really run a programming language, you're talking so many thousands of dollars, it was still out of the price range of anyone. It would be like $5,000, and, I'm sorry, but we were all low-level, just barely-getting-along-type people that had this interest in having our own computers.

Secondly, 5 years before that, in 1970, I had built a computer of my own design that was exactly what an Altair was—only I didn't have a microprocessor; I had to build it out of chips. So I built a little processor and it was only on one small—almost 3-by-5—card, very tiny. It had switches, it had lights, it looked like an airplane cockpit, just like the Altair. It had just as much memory as the Altair (256 bytes was the starting amount of memory). I could toggle these switches, punch some buttons, get ones and zeros into memory and run it as a program, and I could verify it really was in there and running. So I had done this 5 years before. Now I saw the Altair and I saw the microprocessors and I knew that they weren't enough. You needed something to run a whole computer language. But it was close.

So I searched around. My thinking was always, in making something possible, you've got to get it down to a reasonable cost, but I needed 4K bytes of RAM minimum. The first dynamic RAMs got introduced that year, 1975—the first 4K dynamic RAMs. That was the first time ever that RAMs were lower in price than magnetic core memories, which every computer up to that day had used. So all of a sudden, the world was going to change to RAMs. Silicon was going to be our memory.

Everybody else in the world—the Altair, the Sphere computers, the Polymorphic computers, the Insight computers—every one was designed by basically insufficient engineers, not top-quality engineers. They were designed by technicians who knew how to look at the datasheets for some RAM, look at the datasheets for a microprocessor and see if the microprocessor had some lines called "address"—and the RAMs had lines called "address," and they would hook a wire from one to the other. It's a very simple job—if your RAMs are static RAMs.

The dynamic RAMs were going to be one-half to one-quarter the price. The dynamic RAMs meant that instead of 32 chips to have enough memory for a computer to have a language, you only needed 8 chips of RAMs. But dynamic RAM needs all this circuitry to get into every single address in the RAM every 2000th of a second, read what was there and write it back, or it forgets it. Dynamic RAM (this is what we have in our computers today) will forget every single bit in a 2000th of a second unless something reads it and writes it back the way it was to hold its state. It's like little electrons stored on a plate and they'll leak off in a 2000th of a second.

Well, that took some extra circuits and thinking on my part, but when I put my computer together, good lord, I already had these counters that were counting regular sequences for a TV screen, for my terminal, and I said, "I'll just use those counters to supply the counts to sneak in every so often and update part of the RAM." So constantly the microprocessor would get to my RAM and the video addresses would get to my RAM—not to really read video (video wasn't in the RAM back then because I was using the same terminal that I had built before and it had its own memory for the screen), but it would get in and just sample things in the right sequence to make sure the RAM stayed alive. It took a little more designing, but in the end it was a lot less chips. It was not only a lot less chips, but it was smaller in size. It was more impressive to anyone who saw it. It was cheaper and it was faster. You get all these things at once if you use the right approaches.

In the late 1960s, a ton of minicomputers were coming out, and they all used the same chips: 7400 chips that would have 4 gates on a chip—or they'd have an adder on a chip or a quad adder on a chip or a multiplexer on a chip. They'd all use the same chips in all these computers, but what they did was say, "Let's build a computer. Like all the computers before, it has an instruction that can add 1 to an accumulator, has this many registers, it can move a register to memory, it can add, it can exclusive-or them, it can exclusive-or them with memory." They make up an instruction set that will make this computer usable. It will grow into an operating system, it will grow into programming languages, if we design enough instructions into the machine.

Then Data General came up with the Nova minicomputer and, instead of having 50 instructions to do various types of mathematical type things, they had 1 instruction; 1 instruction of 16 bits—6 ones and zeros. A couple of those ones and zeros told it which of four registers to put on one side of the arithmetic unit. A couple more bits told it which other of the four registers to use. Another couple of bits told it whether to shift or rotate the result after it finished, left or right, which is equivalent to multiplying or dividing by 2. There were bits as to whether you should set a carry (just like you learned addition in elementary school, you have carries—well, computer circuits worked the same way). By the time you were done, all of these 16 bits had certain meanings. I looked at it when I went to design a Nova, and it turned out that two of the bits selected one of the four registers, so I ran them to a four-way multiplexer chip and it just flowed in. It's like those two bits fit a chip. I didn't have to make up a bunch of logic that decides "do this and this and this, and gate those over here, and put a signal down there." I didn't have to do all that stuff. It just flowed logically. Three of the bits flowed down to a logic chip to tell it whether to add, or, or exclusive-or. Another bit just got fed in as the carry into the adder. By the time I was done, the design of the Nova was half as many chips as all of the other minicomputers from Varian, Digital Equipment Corp., Hewlett-Packard—all of the minicomputers of the time (I was designing them all). And I saw that Nova was half as many chips and just as good a computer. What was different? The architecture was really an architecture that just fit right to the very fewest chips.

My whole life was basically trying to optimize things. You don't just save parts, but every time you save parts you save on complexity and reliability, the amount of time it takes to understand something. And how good you can build it without errors and bugs and flaws.

**Livingston:** You were designing all of these different types of computers during high school at home, for fun?

**Wozniak:** Yes, because I could never build one. Not only that, but I would design one and design it over and over and over—each one of the computers—because new chips would come out. I would take the new chips and redesign some computer I'd done before because I'd come up with a clever idea about how I could save two more chips. "I'll do it in 42 chips instead of 44 chips."

The reason I did that was because I had no money. I could never build one. Chips back then were . . . like I said, to buy a computer built, it was like a down payment on a good house. So, because I could never build one, all I could do was design them on paper and try to get better and better and better. I was competing with myself. But that's just the story of how my skill got so good. It's because I could never build anything, I just competed with myself to come up with ideas that nobody else would come up with.

I knew that I had a lot of approaches in computers that basically no human really would use. They couldn't even be taught in a school program. I did a lot of it in my head. Taught myself everything. We didn't have computers in our high school even. And I was designing them. So, I just came across some lucky journals and then I discovered a way to get computer manuals. The computer manuals described the computers and my dad got me chip manuals. So I just figured out, "How do you take the chips and build a computer?"

My skill was that, if I know what I want for the end result—in those days it was a computer, in later days it might be a certain floppy disk that had to read and write some data—but if I knew what my end goal was, I know how to combine chips together very efficiently to get that goal done. Even if I've never designed anything before. My skills weren't that I knew how to design a floppy disk, I knew how to design a printer interface, I knew how to design a modem interface; it was that, when the time came and I had to get one done, I would design my own, fresh, without knowing how other people do it. That was another thing that made me very good. All the best things that I did at Apple came from (a) not having money, and (b) not having done it before, ever. Every single thing that we came out with that was really great, I'd never once done that thing in my life.

**Livingston:** Do you think that that's a recipe for being good at something: you've never done it before and you are trying to do it on the cheap?

**Wozniak:** Yup. But you have to have skills. We had a guy that designed the Macintosh and he was the same way. He'd never gone to college, but, boy, he just studied circuits that had been done by others and just became that good on his own.

**Livingston:** You went to college and then dropped out, right?

**Wozniak:** Not exactly. But I didn't learn anything about designing computers in college. I never had a class, for example, in writing a computer language, and, when I got my computer done, I had to write a Basic. It needed a Basic, there was no other choice. I also knew how to combine low-level software to build a program that was immense. I didn't know anything about computer languages except—a friend of mine had gone to MIT and, while he was there, he would Xerox pages out of books that were good topics, and he had sent me a lot of pages back from compiler design books. So I had actually read some compiler design books. I hadn't taken a course, I hadn't had a teacher, but I had some ideas of some of the parts involved in parsing a computer language.

So when I got my computer built, the Apple I, I just took the terminal that I already had. It was a shortcut computer; it was not designed to be an efficient computer from the ground up—that was the Apple II. This one was: take the terminal that I already have that works on my TV set and has a keyboard. And then I said, "All these computers are coming out and they've got switches and lights and look like airplane cockpits, and they're just like the one that I built 5 years before"—Cream Soda Computer we called it. And I said, "That was just too slow and sloppy. It was neat to have a computer, but it didn't do what I wanted to do. I want to write a program in Basic; I want to type in a game and play it; I want to write a program that solves my simulations for my work at Hewlett-Packard." (I used their big computer. They had a minicomputer that was shared by 40 engineers so you'd sign up for time on it.)

I knew that I wanted a good enough computer and it meant a microprocessor (once I discovered that a microprocessor was like those minicomputers I used to design), dynamic RAM was the choice to save money and parts, and I already had the terminal. Then I sniffed the wind and I said, "I need a language. I've got a 4K computer. It can run a language, but there's no language yet for this microprocessor. So I was (a) a little bit disappointed because I wanted a computer language, but (b) I was excited and exuberant because I got to be the one to write the first language for this processor. I would get a little bit of fame out of that, and I was super shy, so the only way I could ever get noticed was if I designed great things.

So I got to write a computer language, but remember I've never written one in my life. I'd never taken a course on it. So I opened up the Hewlett-Packard manual at work and saw the Basic. I read all the different commands in the Basic, and I started creating a syntax table that showed the grammar of that language: what words, what commands are allowed in what order, how you put in variable names, how you put in numbers, what size they can be, what formats. Then I came up with an idea—and I have no idea where it came from—just a weird, weird idea that, as a user types in a statement, I will just scan his statement, character by character, from left to right, and I'll see where it fits into the syntax table. I typed my whole syntax table into memory. I said, "I'll just follow along in memory and, if what he types fits the syntax table, then whenever he hits return, I know all the elements he typed in." I just output a list of little tokens that represented what had been typed in, if it matched the table. This was just an idea I had, not knowing how other people did it. I don't know to this day how compilers are written.

I also knew that there were numbers and variables and you have operations like plus and minus, times, divide. (I was just a very low-level person here . . .) Numbers are nouns and a plus is a verb. Even in a statement like "print," print becomes a verb. So I had these lists of verbs and I had noun stacks and verb stacks and figured out ways to push them on and make their priorities such that we could turn it into reverse Polish notation.

I was very familiar with reverse Polish notation from books I read in college (or that my friend had sent me in Xerox form); and also our Hewlett-Packard calculators used reverse Polish notation, and we thought we were more advanced because we were doing what computer science people do. You take an equation like "5 + 4" and you change it into "5 ENTER 4 +" so you do the addition last. But how do you convert between one and the other? That one wasn't too bad for me. I had some knowledge of that.

I built this whole Basic up and it worked, and that was the hardest project I did. Normally you type a computer program into a computer; that's the only way it's done. You type it into a computer or you feed it in on cards. What I did was I handwrote it on the left side of the pages in my program, in what's called machine language. That's as close as you can get to the ones and zeros. And then I looked at a little card and I translated my program into ones and zeros on the other side. If it said, "Jump ahead," I'd have to count—if it's jumping ahead 19 bytes, I'd have to write 19 in zeros and ones. I would write the zeros and ones myself because I couldn't afford a computer program that did this assembly job. I went down to the absolute lowest-level jobs you could do. For the computer itself, I not only designed it on paper (I was the draftsperson, I would draft it on my drafting board), I would hook up all the parts and figure out where to plug them into some boards, and I would solder wires between each one.

In my minimalist approach, I made the wires the shortest, straightest, thinnest wires possible, instead of having these big old looped-up hairy messes of wire-wrap type stuff. So I did all that and I was also the technician. I would test things out and look for the voltages first and apply it carefully and look for signals and analyze what was wrong and fix the bugs and resolder and come up with new ideas and add some chips in. I was the technician and everything for all of the Apple projects I ever did.

**Livingston:** So where were you when you first realized that you could build the Apple I?

**Wozniak:** I got this idea that I was going to have the computer that I had wanted my whole life at the first meeting of the Homebrew Computer Club. That night, I realized it, when I found out what a microprocessor was. I went home and studied it and said, "Oh my god, I'm here. Because now I can come up with the money to buy it someday." At first it was quite a job to come up with the money because the Intel processor was $400, and I just wasn't going to come up with that soon. It's like coming up with $2,000 nowadays. That's a big deal. Then I found out there was a Motorola one I could get for $40 at Hewlett-Packard and then the company introduced the 6502 for $20, so that's what I bought. I bought it because it was just super-cheap and it was also the best one of the day.

Now I had to build the hardware. I looked at all the other computers that were around me and they were like the standard old computer—switches and lights and slots to plug boards in and connect them to teletypes. I said, "No, I want the whole thing, because it's affordable now." I've got my terminal and my terminal already has a keyboard for typing on. It's kind of like our Hewlett-Packard calculators have human buttons—a human can understand what they are doing. None of this zero-and-one stuff. So I said, "But the trouble is you have to get programs into memory." I'm starting out with a microprocessor that didn't even have a programming language, so you've got to still stick some zeros and ones into memory. I said, "Why don't I write a simple little program"—a 256-byte program that took two chips to store. And my program read what you typed on the keyboard and did the stuff the front panel would have done, but did it at 100x the speed in the end. And it could also display on the TV screen what was in memory. It could let you enter stuff into memory, and it could run a program at a certain address. And that allowed me to develop further to start typing my ones and zeros. As I developed Basic, I would type the ones and zeros in by hand, and it got up to where I would type for 40 minutes to get my whole program into memory. I would type not ones and zeros, but base 16 actually, get the program into memory and test out bits of it at a time, and see what's going on. So this was not at all a normal project where you have tools. I had no tools; my approach in life was to just use my own knowledge. I know what's going on better if I'm not going through a tool.

**Livingston:** You had your Sears TV and a tape cassette for data storage, right?

**Wozniak:** Yes. Once I got that much of the Basic done, we had to store a big program efficiently somehow on mass media. I used a tape recorder so I wouldn't have to type it in for 40 minutes. But that came pretty late in the game. I had developed the whole Basic without it really.

**Livingston:** And you showed it off at the Homebrew Computer Club?

**Wozniak:** Every 2 weeks I brought my computer, which became the Apple I, down. We hadn't decided to start a company. Because companies weren't my thing, technology was. I'd bring it down and show it to people, and I brought schematics. I'd make Xeroxes at work of all my schematics and pass them out, because—I made sure my name was on it—I was so shy and I thought, "I'll get known by doing good stuff." And I'm telling other people, "You can build your own. This is how easy it is." And I was really trying to say, "You can have a complete computer at a very low price. And not the Altair way." Trying to say that there was a whole different way of computers. Some people got it and some didn't.

**Livingston:** Did the people who got it try to build their own?

**Wozniak:** It was still too much of a job. A lot of them were software people, not hardware solderers. I went over to one young kid's—he was in high school—I went over to his house and helped him wire his own up. I started doing the soldering. A lot of people in the club didn't even know how to solder. It really was more a software group. So not many built it, and that's really where Steve Jobs came in saying, "Let's start a company." He said, "Look, there are a lot of people that want to build it and they can get the chips, but they don't want to solder it all together. So why don't we make a PC board and they can plop their chips in the PC board"—soldering a printed circuit board is easy, there are no wires—"and then they've got it done."

So the idea was that we'd start this company and build PC boards for $20 and sell them for $40. Well, I only knew the club as a place to sell it and I thought, "Are there 50 people at the club"—I had a group gathering around me—"who are going to buy this computer instead of the Intel?" I didn't think so, but Steve said, "Even if we don't get our money back, at least we'll have a company." So it was like two good friends having a company.

**Livingston:** Do you remember where you were when you guys talked about making a company out of this?

**Wozniak:** I don't. I don't remember if he phoned me at work, if I was at his house, if he was visiting me—I can't remember.

**Livingston:** How did you know Steve?

**Wozniak:** That computer that was like the Altair that I'd built 5 years before—Cream Soda Computer—I'd told a friend down the block, Bill Fernandez, about it, and we agreed to solder it up in his garage. We spent about 2 weeks soldering my design together. We'd ride our bikes down to buy cream soda and come back and drink it, so we called it the Cream Soda Computer. Bill went to our high school, and he said, "There's another guy at Homestead High School, younger than you, and he's interested in electronics and pranks and things too and you really should meet him." So he thought we were alike.

The way I remember it is that Steve came right out there in front of his house. We're out there on the cul-de-sac on the sidewalk and we're just talking. We started out by comparing pranks we'd done and talking about different types of electronics and chips. We both had a lot of similar experiences so we had a lot to talk about. Then we became best friends for so long. There weren't that many people that young that knew technology. Steve and I weren't similar personalities, which was strange, but I'm the sort of person that goes along with anyone that wants to talk technology. And then we both agreed on music too. We had very strong music influences in those days, and it was more songs about living and life and where we're going and where we're from and what's it all about and what works and what doesn't. It was a lot more Bob Dylan stuff than normal popular music that intrigued us. So we'd go to concerts. I was going off to Berkeley, but I'd be down on weekends. Every time I was down, we'd link up, have a pizza, whatever.

**Livingston:** What were the first things you did after Steve suggested starting a company? You were still working at HP, right?

**Wozniak:** The very first thought in my mind was, "I think I signed a document that everything I design belongs to Hewlett-Packard." Even just on my own time, I thought that they deserved it first. And I wanted Hewlett-Packard to build this. I loved my division. I was going to work there for life. It was the calculator division; it was the right division to move into this kind of a computer.

I went to management, and I had three levels of bosses above me in a room and a couple of other engineers, and I presented the ideas and told them what we could do at what price and how it would work. They were intrigued by it, but they couldn't justify it as a Hewlett-Packard product for some good reasons. Hewlett-Packard couldn't do a simple project, which was really what was interesting. They had to do a real finished-for-scientists type of computer that would be too expensive and really wouldn't start the mass movement. They were a little concerned about using a TV set that didn't come from Hewlett-Packard. When there's a problem, how do you decide where the solution is? But I know they were intrigued by it quite a bit. That was when we were going to sell PC boards for $40 each.

When Steve called me one day at work and he said he got an order for $50,000—100 built computer boards for $500 each—that was high money. That was twice my annual salary at Hewlett-Packard. So then I got Hewlett-Packard's legal department to search every division—I wrote down what we were doing and had them search every division—but the thing is that the calculator division was the lowest one in Hewlett-Packard. The others wouldn't want to touch anything cheap. It was too cheap for our division, and the other ones wouldn't touch it even more. So I got a written response back from them that no divisions were interested.

Now it was almost like we were big-time. We were going to sell some computers. Sure, we only sold 150 (maybe less) of the Apple Is, but it was a real computer and we had our name in all the magazines with charts and comparisons. This whole industry's springing up and there are articles about it. And no article could skip a company with a name like Apple.

**Livingston:** How'd you come up with "Apple"?

**Wozniak:** Steve came up with it. I do remember that one. I picked him up at the San Francisco airport and I was driving down the Bay on 101 and then on 85, and it was on 85 that he said, "Oh, I've got a name for the company. Apple Computer." Both of us were sitting there trying to come up with techie names that were clever, but nothing was going to be better than Apple. And I said, "But what about Apple Records?" (Which is funny because we're still having problems with them.) And he said, "They're a different company."

So we said, "OK, we'll do Apple Computer." In those days there was no money yet in this microcomputer business, and big experienced companies and investors, analysts—those kind of people, that are trained in business and much smarter than we were—they didn't think that this was going to be a real big market. They thought it was going to be a little hobby thing, like home robots or ham radios, that a few techie people would get into and really it wasn't going to go to the masses.

In the Homebrew Computer Club, we felt it was going to affect every home in the country. But we felt it for the wrong reasons. We felt that everybody was technical enough to really use it and write their own programs and solve their problems that way. Even when we started Apple, we had very mistaken ideas about where the market was going to be that big. We didn't foresee the VisiCalc spreadsheet.

**Livingston:** Had you quit Hewlett-Packard?

**Wozniak:** That was very tough. We started selling the Apple Is, and I stayed at Hewlett-Packard. I still intended to be at that company forever. Our calculator division moved up to Corvallis, Oregon, and my wife didn't want to move to Corvallis and I did, so that was lucky because otherwise I would have been up in Oregon and Apple never would have happened. So I stayed here and I moved into another division of Hewlett-Packard across the street that made the Hewlett-Packard 3000 minicomputers.

I was working there for a while getting educated on the HP 3000 . . . for the Apple II, we knew it was so good . . . that was a product that broke ground in every which way. The Apple I, oddly enough, was probably more important, because it said that a computer of the future is going to have a keyboard and a video display and it's going to look like a typewriter. It's going to be roughly that size. And it's funny, but every computer since the Apple I, including the Polymorphics technology Sol computer that came next (it was out of our club), had a keyboard and a video display. No computer had done this before that. No small computer was coming with a keyboard yet. The Apple I was the first and the Apple II was the third. Basically every computer since then had a keyboard and a video display. The world has never gone back from that day. Now the Apple II was the great design. I designed it very efficiently with very few parts—amazing design. We added color. How could you ever have color and still cut the chips in half? It was half the chips of an Apple I. It had color, and it was just a clever idea that popped in my head one late night at Atari.

When you get very, very tired—and I had been up four nights all night long; Steve and I got mononucleosis—your head gets in this real creative state and it thinks of ideas that you'd normally just throw out. I came up with this idea of taking one little cheap (less than $1) part with 4 bits in it. If I spun it around at the right rate, the data that comes out of that chip looks like color TV. And I could put 16 different patterns and they all look like different colors, sort of. Would a digital signal that goes up and down actually work on a color TV the way there are sine waves and complicated calculus to develop how color TV was established in the television world? Would it work?

Man, when I actually finally put together this little circuit and put some data into memory that should show up as color and it showed up color, it was just one of those eureka moments and you're just shaking inside. It was just unbelievable. Here we had it in just a couple of chips. I had color, and then I had graphics, and then I had hi-res, and then I had paddles and sound to put games into the machine. It had dynamic memory—it had the newest right type of dynamic memory that could expand almost forever. All sorts of slots with a little mini–operating system that actually worked incredibly well. The Apple II was just one of those designs. Anybody could build things to add on to it, anybody could write programs, they could write sophisticated programs, they could write it in machine language, they could write it in my Basic. So that machine, there was just nothing stopping it.

We knew we'd sell 1,000 a month, but we couldn't afford to build them. So we sought money, and one of the first places we went to was Commodore. To the guy who had been the product marketing manager for the 6502 microprocessor that I had chosen. I had actually bought them at a show in San Francisco over the counter for 20 bills. He and his wife would hand them to us at the table. That's how we bought our first microprocessors that became the Apple I and Apple II, from this guy Chuck Peddle. He now was moving to Commodore to do a computer. We said, "We've got to show him the Apple II."

So we brought him by the garage. I really respected the guy; he designed the microprocessor that I had chosen. He came to the garage and looked at the Apple II, and I put it through all its specs of bringing up quick patterns on the screen and scrolling text and playing games—all the things I'd done on it. He looked at it and didn't say too much. I figured he'd be more impressed. We later heard that Commodore turned it down.

We went in and spoke one day to Commodore's head of engineering, Andre Sousan, and Andre told us that his boss who ran Commodore, Jack Tramiel, had basically brought in Chuck Peddle and Chuck had talked him into "No, you don't want to put all these exotic things like color into it." The truth is, he didn't know how to. No one knew how to do color cheap. There were boards out for small computers. Cromemco had a color system. You buy two boards for your Altair; each of those had more chips than the Apple II on it. So, just to add color, that's what it was like for most people. And Chuck Peddle said, "You should do it cheap. We should just have black and white; we should have the cheapest keyboard you can imagine, the smallest screen, and just keep the costs way down." They wanted to make it cheap enough to be affordable. The funny thing is that the Apple II had so few parts, it was cheaper to build and still was much more of a computer. We didn't have to include a TV set, because we assumed everyone had their own.

**Livingston:** Why didn't Commodore want it?

**Wozniak:** Good question. Andre Sousan very soon after (within weeks) left Commodore and came to Apple saying that he felt we had the right product and he wanted to be with us. They just missed the boat. I think it was that Chuck Peddle knew what he could design, but he knew that he couldn't design what the Apple II was. They should have bought it. They would have had a real good deal cheap. After that, we were still seeking money. I wasn't really seeking the money, Steve Jobs was. I mean, I almost couldn't have cared less. If I could show it off at the club and get credit for having a great computer design in my life, that's what I wanted. We went down to visit some Atari friends. We went to Al Alcorn's house, and he had a projection TV—the first time I ever saw a projection TV in my life, really. And we put it on his projection TV and he looked at it and he liked what we were doing. He was real interested. Atari would do this, but they had a hot project coming out—the first home Pong game—and they were going to have so many millions of those that every effort in their company had to go that way. They didn't have the ability to do two things at once. So they turned us down, very friendly though.

Then we talked to some venture capitalists. Don Valentine came to the garage and he looked it over and he didn't seem too impressed. He would ask questions like, "What's the market?" And I'd say, "A million." And he'd say, "How do you know?" And I said, "Well, there's a million ham radio operators, and computers are more popular than ham radio." Nobody in the world could ever deny that. But it's not the sort of analysis that they wanted. And there were no analysts yet that were predicting that this was going to be a big marketplace anyway.

So Don wasn't that interested, but he gave us the name of Mike Markkula—Mike being a person who was interested in technology, who was looking around for things to do. So Steve went over and talked to him and Mike really thought we had a great thing, that there was going to be a huge market for small computers in the home. Home computers. We didn't even have the word "personal computer" yet; that came about a little later. Because we were trying to say, "How do we establish this new type of computer? What's special about it?" In the old days, several people would use one computer all at the same time. This was the first time you'd have one computer all your own. So it's a personal computer. It's almost maybe a negative in some ways, but we're making it a positive.

So Mike said that he would put in the money we needed to make 1,000 computers—$250,000. Boy, that sounded astounding. $250,000 back in those days was like a couple million today, maybe.

**Livingston:** Were you still in Jobs's parents' garage?

**Wozniak:** Well, actually we never did much in the garage. People think we had a garage where we sat down with soldering irons and we designed stuff. No. The only designs that ever took place in the Apple I or II for hardware or software were in my apartment in Cupertino or my cubicle at Hewlett-Packard late at night. That's the only place any building got done.

The computers were manufactured at a place in Santa Clara. They made the PC boards, they stuffed the parts in, they wave-soldered it. Steve would drive down and then drive them back to his garage. We did use the garage at his place—we had a lab bench there and we would plug in the PC boards of the Apple Is and test them on a keyboard. If they worked, we'd put them in a box. If they didn't work, we'd fix them and put them in a box. Eventually, Steve would drive the boxes down to the Byte Shop in Mountain View or wherever and get paid, in cash. We had the parts on credit and we got paid in cash. That was the only way we could do the Apple Is.

**Livingston:** So you'd keep self-funding?

**Wozniak:** Yes, we kept self-funding and we probably built up a bank account of about $10,000. Not a huge amount, but it was enough to move into an office. Steve really wanted to make a company.

**Livingston:** Where was the first office?

**Wozniak:** The first office was even before we worked a deal with Mike Markkula. We arranged to get a place at an office complex I could drive to in Cupertino. It's not too far from where Apple's places are now. Not too far from where our first building on Brandley was. We had one office and Steve had arranged that we only pay for half of it until a certain date when we'd use the rest. It was kind of cold and empty when we finally did move in.

So Mike was going to finance us, and then one day he said to me, "You have to leave Hewlett-Packard." And I said, "Why? I designed two computers and cassette tape interfaces and printer interfaces and serial ports and I wrote a Basic and all this application software, I wrote demos, and I did all this moonlighting, all in a year."

He said, "Well, you have to leave Hewlett-Packard." It just wasn't open. I went inside of myself and thought about it. "Who are you? What do you want out of life?" And I really wanted a job as an engineer forever at a great company (which was Hewlett-Packard). I wanted to design computers and show them off and make software. And I can do that on my own time. I don't need a company to do it. So there was an ultimatum day—I had to decide by a certain day if I was willing to do this. I met Mike and Steve at Mike's cabaña at his house in Cupertino. Eventually we got around to it, and I said, "I've decided not to do it, here are my reasons." Mike just said, "OK." Steve was a little more upset.

About the next day after I said no to starting Apple, my parents called me and said, "You really ought to do this." (Because $250,000 was a big deal in anyone's life.) And then friends would start calling me. That day my friend Allen Baum called me in the afternoon, and he said, "Look, you can start Apple and go into management and get rich, or you can start Apple and stay an engineer and get rich." As soon as he said it was OK to do engineering, that really freed me up. My psychological block was really that I didn't want to start a company. Because I was just afraid. In business and politics, I wasn't going to be a real strong participant. I wasn't going to tell other people how to do things. I wasn't going to run things ever in my life. I was a non-political person and I was a very non-forceful person. It dated back to a lot of things that happened during the Vietnam War. But I just couldn't run a company.

But then one person said I could be an engineer. That was all I needed to know, that "OK, I'll start this company and I'll just be an engineer." To this day, I'm still on the org chart, on the bottom of the org chart—never once been anything but an engineer who works.

**Livingston:** So you called Steve?

**Wozniak:** I made my decision by that evening and I called Steve and told him I would. Then the next day I came in (to Hewlett-Packard) and I told a couple of friends, who had come over with me from the calculator division. I told them that I was going to leave Hewlett-Packard and then I went over to tell my boss, and he wasn't there. He was in a meeting or something. All day long people started coming up to me saying, "I hear you're leaving." And my boss hadn't heard. Finally he showed up at his desk, and I went over and I told him that I was going to leave and start Apple. He said, "When do you want to go?" and I said, "Right now." So I left that day and the deal with Mike Markkula was that I'd have the same salary starting Apple. It was like $24,000 a year.

**Livingston:** Did you go straight over to Apple?

**Wozniak:** I walked out that day. We didn't have an office yet so I was still at home, but I was doing the Apple stuff. I was finishing up things on the Basic, finishing up some hardware things, writing code for some special graphics, that sort of stuff. Then Steve and I met a friend of Mike Markkula's named Mike Scott, and we liked him very much as a strong, forceful guy (he was a director at National) who got things done that needed doing. We decided that we wanted him to be our President. He was our President from the day we started Apple as a real corporation—until the day we went public, he was still our President. So he had a rather important role in history, and he's very much forgotten. I just think that he was the greatest thing ever.

**Livingston:** How did you find him?

**Wozniak:** Mike Markkula knew him as a friend. Their friendship kind of came to a breaking point where Mike Markkula sort of ousted him as President for making rash decisions. There was a day that he laid a lot of people off. Apple kind of grew and grew and grew and had a bunch of engineers assigned to different projects, and we weren't getting out really good stuff really fast like we had been. Mike Scott came in and told our engineering manager, Tom Whitney (a guy that I worked for three times in my life: once at Hewlett-Packard's calculator division, later on at the Hewlett-Packard 3000 division, and now at Apple), to take a vacation for one week, and he went around and talked to all the engineers and found out who was doing stuff and who was slacking off. He pretty much fired the right ones—that weren't working. But he should have given them chances to go around and bring their abilities to play and all that.

Mike Markkula was close to Ann Bowers at the time (she was the wife of Robert Noyce, I think), and she was taking over our human resources. So to have this poor of an example of human resources was almost a blot on the face of the company. Mike Scott was starting to make some real rash, quick decisions, and not be as careful as was needed, and as he'd been in the past. The board gave him another job and he wrote a very shocking resignation letter that, basically, life was too important for this political type stuff. It was sad to see him go because he supported good people so well in the company.

**Livingston:** What about Ron Wayne? Wasn't he one of the founders?

**Wozniak:** Yes, but not when we incorporated as a real company. We had two phases. One was as a partnership with Steve Jobs for the Apple I, and then for the Apple II, we became a corporation, Apple Computer, Incorporated.

Steve knew Ron at Atari and liked him. Ron was a super-conservative guy. I didn't know anything about politics of any sort; I avoided it. But he had read all these right-wing books like *None Dare Call it Treason*, and he could rattle the stuff off. I didn't realize it until later.

He had instant answers to everything. He had experience with businesses and times he'd been gypped out of stock deals. He always had something very quick to say and, wow, it sounded like he was very knowledgeable about this stuff. He sat down at a typewriter and typed our partnership contract right out of his head using lawyer-type words. I just thought, "How do you know what to say, all rights and privileges and all the different words that are in there"—I don't even know what they are. He did an etching of Newton under the apple tree for the cover of our Apple I manual. He wrote the manual. So he helped in a number of ways. Steve had 45 percent of this partnership, I had 45 percent, and Ron had 10 percent, because both of us agreed that we could trust him to resolve any dispute, and we would trust his judgment.

Then what happened was that we were going to sell PC boards for $20 each and fund it out of our own pockets. I sold my HP calculator, Steve sold his van, so we had a few hundred bucks each. Then Steve got the $50,000 order. Over at the company that was making our PC board, as soon as the PC boards were made, they opened up a closet that had our parts and it started a 30-day clock ticking. We had 30 days to pay for the parts. The parts got stuffed into the computers, we made them work, we delivered them to the store and got paid in cash. The parts suppliers—the distributors in Mountain View—had checked with the store owner and knew that he was going to pay us. So basically, we didn't have the credit; he was good for it. But, here was the problem: What if he didn't accept them one time or didn't pay us? We would owe a ton of money on those chips.

I had no money and Steve had no money. We didn't own cars, we didn't have savings accounts, we didn't have houses. So Ron Wayne figured they'd come after him for his golden nuggets that he kept under his mattress. (He actually tells me it was in a safe—but he was afraid they'd come and get his gold.) So he sold out. It was too risky for him, so he sold out his 10 percent of Apple to us for a few hundred bucks. Maybe $600, maybe $800, maybe $300—but a few hundred bucks. And this was even when we had an Apple II designed and were heading toward future business. He was just scared that something was going to catch him.

**Livingston:** Way back then, how did you guys divide the work between you?

**Wozniak:** We actually never talked about it even once. If there was any engineering to do, hardware or software, I did it, because Steve could do stuff, but he couldn't do it as well as I. So never once did he even try. Never did he look at a circuit and suggest anything. I don't want to mess around running a company—my whole life's engineering—so he's on the phone talking to reporters, talking to stores, "Do you want us to ship you some computers, do you want to start buying them?" Talking to the dealers on the parts, ordering the parts, negotiating process, getting brochures made up or ads for magazines.

**Livingston:** So you two fit together nicely in terms of your skills.

**Wozniak:** Well, we added up to the total everything that was needed. If there was anything that neither one of us knew how to do, Steve would do it. He'd just find a way to do it. He was just gung ho and pressing for this company to be successful. And me, I was pretty much only in my technical head with the circuits.

**Livingston:** Do you remember any disagreements you had in the early days?

**Wozniak:** Extremely minor. There were a couple, maybe. One was that we're getting close to shipping it and we wanted things to be low-cost. Steve says, "Can we save any chips?" He's pressing me and pressing me. I am down to like what is just amazing in the world. People to this day that understand circuitry tell me how they looked at my design and it was the most beautiful thing they ever saw. So I said, "I could cut out two chips if I skipped high-res. I don't know if anybody's really going to use high-res." (It became very important actually.) And Steve said, "Oh no, if it's only two chips, leave it in." But it wasn't like we were really arguing. I was just telling him that that's the only place I could save any chips.

We had a real argument over slots. Mike Markkula's coming on and we were going to build the Apple II, and I had designed a clever system on the suggestion of a friend—Allen Baum again—that decoded eight slots you could plug little computer boards into. Each board had the ability to have its own programs on it running in its own addresses, and it didn't have to have all the normal chips to decide, "Well, if the addresses are such and such, I will respond to them." That was done on the main board. In the Altair world, each board you had to dial in the address that it would look at, and that took a couple of thumbwheel switches to dial the address on (they cost money), and a bunch of chips that would compare the address coming from the microprocessor to the one that they were good for, to see if they equaled, and that cost about 5 chips a board. So if you had 8 boards, that would be 40 chips. In my case, I used 2 chips, and I had double sets of address to all 8 boards already in 2 chips instead of 40. So I was very proud of that.

Now Steve said, "All people really need is a printer and a modem." And that was just false because he'd come from a different world than I. He'd never done software and he'd never really been around computer users. He'd been around Hewlett-Packard where they make them, but he hadn't been around computer users that plug in boards that do an oscilloscope out of a computer board, and another board that controls some equipment on the factory and runs some motors, and all these little boards that were just a big part of my life. Every computer I'd ever seen, some of its greatest things came because of boards plugged into it. And he wanted just one slot for a printer and one for a modem. Today, we're sort of in a much different, freer world.

We got the computer finished up enough. We don't have much to add on besides a printer and a telecommunications of some sort. So Steve was arguing for two slots. And the trouble is, two slots wouldn't save me a single chip. And I wanted to show off that I had eight slots and so few chips. If I only had two slots, I would have had parts of chips unused. I was really dead set to hold my chip count, so I said, "If you want two slots, get another computer." That was the only time we had a real argument.

**Livingston:** Did he keep pushing?

**Wozniak:** No, he had no choice. I gave him no choice. We had to have eight slots. And it turns out that it was very important; it was very beneficial. Because we came out with a floppy disk. Not only that, other people came out with cards that put 80 columns of text on the screen so you could see more. People came out with extra memory cards, people came out with other languages in cards, people came out with cards that had CPM. People came out with cards to connect all kinds of equipment in the world, to operate your house over your power lines. It was just a world of cards. Many people had their Apple IIs filled up with cards—every single slot.

**Livingston:** When you showed people the Apple computer, were they amazed?

**Wozniak:** Every single time I showed the Apple II, before we started the company and even slightly after we started the company—before there was much word around about it, every single person who ever saw it . . . The engineers at Hewlett-Packard came to me and said, "That's the best product I've ever seen." And they're around one of the greatest products of all time—the Hewlett-Packard calculator—and one of the greatest companies, and they're saying things like that. The Apple II had so much intrigue to me, but I knew it intrigued all technical people. And the Apple I just worked. I actually wound up doing some great work at Hewlett-Packard using that as my computer.

**Livingston:** What is the key to excellence for an engineer?

**Wozniak:** You have to be very diligent. You have to check every little detail. You have to be so careful that you haven't left something out. You have to think harder and deeper than you normally would. It's hard with today's large, huge programs.

I was partly hardware and partly software, but, I'll tell you, I wrote an awful lot of software by hand (I still have the copies that are handwritten), and all of that went into the Apple II. Every byte that went into the Apple II, it had so many different mathematical routines, graphics routines, computer languages, emulators of other machines, ways to slip your code in and out of an emulation mode. It had all these kinds of things and not one bug ever found. Not one bug in the hardware, not one bug in the software. And you just can't find a product like that nowadays. But, you see, I had it so intense in my head, and the reason for that was largely because it was part of me. Everything in there had to be so important to me. This computer was me. And everything had to be as perfect as could be made. And I had a lot going against me because I didn't have a computer to compile my code, my software.

**Livingston:** Did you have a hard time getting everyday people to say, "Yeah, I want a computer in my office, my home"?

**Wozniak:** Almost everyone who saw it wanted one, but usually the idea was, "What's the cost?" A couple thousand bucks. "Well, I want one of those." But they weren't jumping because it's enough money—you have to plan and maybe some months ahead downstream, you'll be able to buy one.

But we never found one person who said, "I wouldn't have any need for this at all." (We didn't talk to elderly people.) But people not only in their offices, but just at home, you play one game on it, and an awful lot of people—adults and children—want a machine to play games. The Apple II really started the whole gaming industry, because it was the first time a computer had been built with sound, paddles, color, graphics—all the things for games. And it was really so that I could implement Breakout in software.

Back a year before, when I had worked at Atari, they were starting to talk about coming out with microprocessor games. Up till then it was all hardware. In other words, you solder wire to the right sort of chips and put it through some more chips and some other chips, and it determines where the score is on the screen. It's not like you type it in software and say "put the score at this location." No, it was all done with wires and gates and chips and registers, and it was very difficult back then.

So now I had a machine that I could program a game in (or somebody could), and I got this crazy idea to try to do Breakout in Basic. Basic is like a hundred to a thousand times slower than machine language, so I don't know if it's possible. I sat down one night and finally put in all the commands in the Basic to draw color, and I started typing away in Basic and, within half an hour, I not only had my Pong game working, but I had done about 50 or so variations of colors and speeds and sizes and where the score was and all that stuff. I had changed so many things around and put in little features that would just take forever to do in hardware. Little words pop up on the screen when things happen. I called Steve over and I was just shaking, I was quivering, and I showed him the game running, and I said, "This game was so easy to write! Look at this, go ahead—change the color of the bricks." This would have taken me a lifetime to do in hardware and I did it in half an hour.

And that was true. It would have taken an entire lifetime for any engineer with a soldering iron to try all those variations. So I said to him, "Now that games are software, it's going to be a different world for games." And the Apple II, so many people just started trying to figure out how can you get rocket ships to launch, how can you get things that sound like sound when you have a real cruddy voltage to a speaker. How do you listen to somebody talk and figure out what they said? They started using the Apple II. It was just open to all these things. We made it easy for anyone to do what they wanted to do. And I think that was one of the biggest keys to its success. We didn't make it a hidden machine that we own—we sell it, it does this, you got it—like Commodore and RadioShack did.

We put out manuals that had just hundreds of pages of listings of code, descriptions of circuits, examples of boards that you would plug in—so that anyone could look at this and say, "Now I know how I would do my own." They could type in the programs on their own Apple II and then see "that's how that works" instantly, and know how to write their own programs. Running cards was the most important thing. All these companies started up making cards that you could plug into your Apple II and write a little software (mostly games at first) on cassette tapes. You'd go to the store and they'd just have all this stuff that you could buy to enhance the Apple II. So one of our big keys to success was that we were very open. There's a big world out there for other people to come and join us.

In the years 1980 to '83, when the Apple II was the largest-selling computer in the world, we didn't advertise it once. Everybody else who was making products for it was advertising for it. All of our ads were for the Apple III, which never sold in that time frame. Because we were trying to make the Apple III the big business machine instead of IBM.

**Livingston:** That didn't happen, right?

**Wozniak:** That didn't happen. I think it was a total fallacy. I think we should have advertised the Apple II. If you've got the world's best-selling computer, keep it going as much as it can. But the company kind of wanted the Apple III to win and the Apple II to lose. It was really weird because you'd walk into the company and everybody had an Apple III on their desk—nobody had an Apple II. The Apple II was the largest-selling computer in the world, and the only guy working for it in the company was the guy reprinting the price list.

Then by '83, the IBM PC took over. It was selling more computers than the Apple II.

**Livingston:** You had left by then, though, so you weren't part of the Apple III, right?

**Wozniak:** I didn't exactly leave. I didn't leave college either; I didn't drop out. Between my second and third year of college, I worked for a year programming to earn money for my third year. After my third year of college, I crashed my car and totaled it. It was a very famous night, the night I met Captain Crunch of blue box fame. Later that night, I got home, picked up my car, drove back to Berkeley at 3:00 a.m., and I fell asleep on the freeway and totaled my car. I walked to my dorm and told my roommates, "It's a good thing I didn't pay the quarterly parking fee."

So after my third year of college, I took a year off to work, to earn money for my fourth year. Then I got that job at Hewlett-Packard. What an incredible job. And then my career started going up, and I had all these side projects that I was working on and then Apple. So I never really had a chance to get back. But I was close, and I wanted to get back. And in 1981, I had a plane crash. As soon as I came out of amnesia from the plane crash—within 5 minutes I knew that this was the time I was going back to college. I'd never get another chance. So I went back and got my degree. I always liked school and was a good student, a top student. And my parents had college degrees and I thought something of that. My kids should see their dad with a college degree.

**Livingston:** Any other eureka moments in the early days?

**Wozniak:** I've told you two major eureka moments. One was getting color to work, with this weird scheme that I had no idea if it's going to work or not. The other was that I didn't know if I was going to get Basic to program an arcade game, and it worked. In both those cases, I didn't even know if it was possible and lucked out. The floppy disk was probably the third real major eureka story.

We had the computer out, and I got to work designing parallel cards to talk to early cheap printers. Then serial cards to talk to better letter-quality printers that are more like the quality work that a business could put out. Then cards that would talk to modems, other serial cards. I actually did a phone card that could control your phone line and control cassette tape recorders and make an answering machine for you and do all this stuff, but it didn't do a modem, just controlled your phone line. Apple never put it out, because they didn't like the guy that I had brought in to do it, which was Captain Crunch. He designed it. It was a great card.

Then came a point where we only had a cassette tape interface at first. To read a program in, you'd stick a cassette tape in a tape recorder and type something on the keyboard and then press a button on the tape recorder. I think on the keyboard you would just type something like "100R" and it means the program goes into address 100. You press the button on the tape recorder and there's a long lead-in period and then data (there's a twiddling sound if you're listening), and you have to wait for a minute and it goes "beep," and now your program is in memory. It worked surprisingly well, but it took a long time.

Mike Markkula wanted to get going right away on the marketing. He ran the marketing for the company. Marketing largely meant, how are you going to present the computer to be acceptable in the home? How do you move "computer" from a word that's yucky and airplane cockpittish to acceptable in my home? And that had to do with different types of photography, pictures, settings, words to the press. He also wanted us to start getting to work on software that would apply.

He basically wanted us to write a flash card program. So Randy Wigginton and I did a flash card program called Color Math and it shipped with every Apple. We also did one called Checkbook, which would let you reconcile your checks on the computer. But here's the problem: you had to first read the Checkbook program in off of a tape, then twiddle your thumbs for a minute and it goes "beep"; then you have to pull out another cassette tape of your own and read your checks in and it goes "beep"; then you have to do the stuff on the screen, enter some more checks and reconcile them; and then you have to put that data cassette back in and record onto it and it goes "beep." You have all these waiting periods, and it was just too awkward and too slow. So Mike said we needed two things: a floating point Basic (that's a Basic with decimal points, which I didn't have) and a floppy disk.

Just before I left Hewlett-Packard, a new chip had come out. The chips in those days were in 14-pin packages and 16-pin packages. This new one was like an 18- or 20-pin package, a little longer than normal, but it had this beautiful little 8-bit chip register, and 8 bits is a magic number—it's a byte. And I had thought, "That chip would be beautiful for getting 8 bits of data off of a computer and shift it out to a cassette tape recorder, or whatever, to a floppy disk. I'd thought about using that chip for a floppy disk, because Steve Jobs had talked about floppies back before I left Hewlett-Packard.

So I said, "I'll look into this floppy disk." And I started pulling up the datasheet on that chip, and I started coming up with my first ideas of "How do I have that chip get the data to a floppy disk?" And then I came up with this clever little approach. I needed a little bit of logic in here, but if you put in logic, you only get four gates on a chip. And you have four gates and four gates and four gates—you need lots of gates to do all this figuring out what to put out, and it's chips and chips. So I said, "Why don't I do a clever little scheme? Data's going to come back from the floppy disk and I'm going to sit there and, within small portions of a microsecond difference, I am going to tell when the signal went from high to low and low to high and tell what the data is."

I needed a little bit of intelligence running at a very high speed, and I came up with a device called a state machine. I'd had a state machine class at Berkeley. I built just a very simple state machine, which basically was a register that contains an address that you're at—a certain place in a program. It held an address as a number and it fed its data into a ROM that took where you are in the program, plus a couple of inputs coming from the floppy disk and from the computer, and decided what it would do next. It would send out signals to cause the right things to happen, and the next address, the next place—it's called a state. So you're in one state and you say, nothing happened, I stay in this state; nothing happened, I stay in this state. Aha, the data from the floppy changed to a 1. I pop down to state number 5 and now I'm in state number 5 and nothing happens, and then the data from the floppy disk just went to a 0, and I pop down here and I also tell a ship register up there to ship in a bit of data, so it actually worked like a small microprocessor even though it was only two chips. It was very successful, a little 256-byte ROM and a little 6-bit register, I think.

So that's three chips, and then I had a couple more interface chips, and I took Shugart's floppy disk. They had a new 5-inch disk, and Steve got me one. Smaller than before—the prior ones were 8-inch. I'd never seen a floppy in my life, by the way. I'd never used or seen one. So I didn't know the first thing about them. I'd never taken a course in floppy controllers, I'd never seen a floppy controller, I didn't know what they did. But I knew on a cassette tape, I generated signals of certain timing patterns and, when they came back from the cassette tape, I analyzed them to figure out what were the ones and what were the zeros. The microprocessor did the timing, because the timing was loose; it wasn't in fractions of a microsecond. I just wrote programs that waited a certain amount of time and saw when the signal went from high to low or low to high, and made decisions right in the microprocessor of our Apple II. But I couldn't do that on the floppy disk. So I looked at Shugart's design to figure out how it worked. And I figured out, oh, you put some data here and some signals here and you set a clock bit at a certain speed every 4 microseconds, and you shipped in some new data. I went through chip after chip after chip on theirs, and I said, "If I take all these out, it's just as easy for me to run the wires straight over to the head that's writing onto the disk. And the signal coming back from it, I just run a wire over to my controller and I just do all the timing here and I don't need all their complicated interface to work." So I took 20 chips off their board; I bypassed 20 of their chips.

Steve Jobs really liked this because, when it came negotiation time, he said, "That's a good reason to sell it to us at a lower price. We don't need your controller board. All we need is a little bit of it. So you can sell it to us cheaper than you are selling it to other people." It was a good deal for Shugart, a good deal for Apple.

I thought I could write some data onto a floppy disk and interpret what was coming back as ones and zeros. Here's the problem: you got a whole big track of data and there's thousands and thousands of ones and zeros and then the track repeats. The head goes around and around. You have to know where and when data starts and stops. And that was an issue I'd never done in my life. I came up with an approach of writing a certain kind of data, a certain pattern—AA D5 AA 55—some pattern like that. I just wrote it for a long enough sequence at the start of every section of data, and it was something that would somehow get my circuits into sync so they knew when a one and a zero started a byte, instead of was in the middle of a byte. It just automatically caused it to just sort of slip into place. By the time it got to the data, it read it correctly. So that was a lucky find. I was afraid, partway through my floppy disk design, that I would never be able to solve that problem. But I did. I lucked out.

Early on in the design, we were going to the very first CES (Consumer Electronics Show) show that was going to allow personal computers—which meant RadioShack, Commodore, and Apple. I had never been to Las Vegas and I wanted to see this beautiful city, but only marketing was going. There was no need for me to go. So I said, "If I get the floppy disk done, then could I go to show it off?" It was 2 weeks away. Something like a floppy disk design, you'd give it 6 months lead time, normally, to write down all the sheets and documents of what you're going to do and get them approved by managers. It's a horribly long cycle. This was 2 weeks away and Mike Markkula said yes. So that was my motivation. I always had these little fictitious motivations that motivated me and got me to do such great work. So I sat down and designed the floppy disk, and Randy Wigginton (he was the guy just out of high school) and I came in every single day including Christmas and New Years for 2 weeks. I came in every single day leading up to, I think it was January 3 or 5, when we went off to Las Vegas. I almost had this floppy disk done.

I got it to where it was writing data on a track, reading the data on a track. Then I got it to where it was reading the data in the right byte positions. Then I got it to work with shifting tracks, and we wanted a simple program where we would say "run Checkbook" or "run Color Math," and it would run the programs that were stored on the floppy disk. So we went off to Las Vegas, and Randy and I worked all night and we got it done to where it was working. At the very end, it was 6:00 a.m. and I said, 'We have to back up this floppy disk." We had one good disk that we prepared with the data hand-massaged to get it just right. So I stuck it in the floppy and wrote a little program, and I typed in some data and I said "read track 0;" stuck in the other floppy and said "write track 0, read track 1, write track 1." There were 36 tracks—I had to switch floppies back and forth.

When I got done, I'm looking at these two floppies that look just the same. And I decided that I might have written onto the good one from the bad, and I did. So I had lost it all. I went back to my hotel room. I slept for a while. I got up about 10:00 a.m. or so. I sat down and, out of my head and my listings, recreated everything, got it working again, and we showed it at the show. It was a huge hit. Everybody was saying, "Oh my god, Apple has a floppy!" It just looked beautiful, plugged into a slot on our computer. We were able to say "run Color Math," and it just runs instantly. It was a change in time.

But the real eureka moment for me was the very first time I ever read data back. I wrote it on the floppy, which was easy—but read it back, got it right. I just died.

**Livingston:** Where were you when you did this?

**Wozniak:** I was actually in Apple's office for the entire floppy disk creation. We were in that office building that I described earlier. There were about five of us in there, then there were about eight or ten. Then I moved out to a second little room that we got—a smaller room in the same office complex but down in another building. Randy Wigginton and I were in there, and Captain Crunch who developed the phone board for me.

**Livingston:** What advice would you give to hackers who are thinking about starting a company or making something on their own?

**Wozniak:** First of all, try to have the highest of ethics and to be open and truthful about things, not hiding. If you have to hide something for company reasons, at least explain what you're doing. Don't mislead people. Know in your heart that you are a good person with good goals because that will carry over to your own self-confidence and your belief in your engineering abilities. Always seek excellence: make your product better than the average person would.

If you can just quickly whip something out and it's done, maybe it's time, once in a while, to think and think and think, "Can I make it better than it is, a little superior?" What it does is not necessarily make the product better in the end, but it brings you closer to the product and your own head understands it better. Your neurons have gone through the code you wrote, or the circuits you designed, have gone through it more times, and it's just a little more solidly in your head, and once in a while you'll wake up and say, "Oh my god, I just realized a bug that's in there, something I hadn't thought of."

Or, if you have to modify something, or add something new, you can do it very quickly when it's all in your head. You don't have to pull out the listing and find out where and maybe make a mistake. You don't make as many mistakes. Just believe that what you have is better than whatever has existed before. We should only move forward in technology and not backwards.

Lack of tools: find a way to do it. If you say, "I have to have a tool," and you are a prima donna—"I have to have a certain development system"—if you can't figure out a way to test something and get it working, I don't think you're the right type of person to be an entrepreneur. Entrepreneurs have to keep adjusting to . . . everything's changing, everything's dynamic, and you get this idea and you get another idea and this doesn't work out and you have to replace it with something else. Time is always critical because somebody might beat you to the punch.

It's better to be young because you can spend a lot more nights, very, very late. Because you have to get things done, and there's almost no other way to get around that. When the times come, they are critical.

**Livingston:** You got mono once because of this?

**Wozniak:** That was the Atari Breakout, because I didn't sleep for 4 days and nights. How could you design a game—this would be months of design—build it, breadboard it, get it working, debug it in 4 days? Steve needed the money quick. He didn't tell me. He also didn't tell me the full amount of the money. He got paid a lot more than he told me, and he only gave me half of a smaller amount. Which he didn't have to; I would have done it for 25 cents. So that wasn't the point. I was glad to just be in there doing it. To get to design a game for Atari, who was bringing arcade games to the world—what a thing to remember for the rest of my life. So I would have done it for 25 cents.

But we both got mononucleosis. There was one Coke can I think we'd shared.

**Livingston:** So he took more money than you did, but you both worked on the project?

**Wozniak:** Yeah, I found out 12 years later.

**Livingston:** That's awful.

**Wozniak:** I know, but he didn't have to. He probably needed the money. And I didn't; I had an engineering job at Hewlett-Packard. It was very little to me. It would have been better if he'd been open about it and honest. And what if I remembered something wrong, too? It's so long ago.

**Livingston:** Did you ever get any investment from Mike Markkula?

**Wozniak:** $250,000. What he did was $80,000 of it was investment for an equal share to Steve and I, and the rest was a loan, paid back to him.

**Livingston:** And that's all Apple ever took?

**Wozniak:** Yeah. But we did right away meet with some people he'd met through Intel that were investment people. Hank Smith of . . . I can't remember the name of the company out of the East, but a venture group. They came in and met us all early on, and they did put in . . . Mike figured out that we were going to need some cash, we were going to be so fast growing. And when you are fast growing, you need more cash right away. So we did have a venture deal in place from well before we shipped an Apple II. And sometime after we were shipping the Apple IIs, we got, I think, $800,000 or $300,000—some large amount—from one venture capital place.

**Livingston:** On the East Coast?

**Wozniak:** I believe that's where we arranged it. Mike Markkula had worked with this guy Hank Smith at Intel, so that's how they knew each other. And I think Don Valentine actually put some money in, but then it came to a point where he wanted to make some good money and buy some stock off Steve Jobs for like $5.50 before we went public. $5.50 a share, and Steve thought it was too low. Oh, those two. Don Valentine doesn't like it when people don't agree with him.

**Livingston:** Is there anything that people have wrong about the early days of Apple?

**Wozniak:** Steve and I never really had an argument. Nobody ever saw us have an argument. The disputes were very rare and minor, of any sort between us, and they were usually just misunderstandings. He'd read something in the paper like I had said it. A lot of times papers got things wrong. They made it sound like I was leaving Apple because I was upset once about things inside of Apple and quoted me on a lot of things. The *Wall Street Journal* did. I told the reporter, "The reason I'm leaving is to start a new startup company to build a remote control. It's something I want to do." I had gone on a whiteboard and shown all the Apple executives what it was so nobody would accuse me of trying to go out and start a company that was competitive. As a matter of fact, they kept me on the payroll. They kept me as an Apple employee. They wished me well and told me that it was non-competitive in writing. But the *Wall Street Journal* got this story down that I was leaving Apple because I didn't like things going on there.

I had complained about the way some Apple II engineers were being treated like they didn't exist in the days of the Macintosh. I mean, we weren't even allowed to buy the floppy disk from Sony that we wanted in the Apple II division, because it would be better than the one that was going to go in the Macintosh. But it was the right one. So that sort of thing. Salaries, bonuses, etc. So I spoke up for some of those engineers in that article, but they made it sound like I was leaving, and I wasn't, not for that reason. Misconceptions . . . there are so many. It's like every book I read that I just think, "God, this is not how this person was at all." So I don't really care, I don't try to correct anything. But the world doesn't really have that much of it wrong in the end. I'm surprised when I go on the Web and I read all sorts of discussions about the Apple II and my role. It's actually very flattering and accurate.

The hardest thing was, though, after having a big success . . . see, I didn't seek the success—I wasn't like the entrepreneur who wants it. So the money to me didn't really mean much. Pretty much I gave it all away to charities, to museums, to children's groups, to everything I could. It almost was like an evil to me. That was because it wasn't the motivation that I was after, and I wanted to remain the person that I would have been without Apple. So that's why I went back and did the teaching. I would have done teaching were there no Apple.

**Livingston:** Didn't you give away your Apple stock early on to other employees?

**Wozniak:** As a matter of fact, when we went public, I was a little disturbed that five people who had been with us in our little office from the start and had been so important—Randy Wigginton, Chris Espinosa, a couple of young kids, and a couple of older ones, just hadn't gotten any stock. I felt that they were a part of this whole energy and excitement and passion for what computers were going to be and what we were doing and how right it was. If somebody is sitting there working till 2:00 a.m. with you, helping to write a little code, and says, "Wow, that is a cool one," those words mean a lot to you and they deserve something. So I gave each of those five a large amount of stock, probably a million dollars in that day. And that was an early day for a million dollars.

I also did a program where I sold stock to about 40 Apple employees . . . I had a chance to sell some stock and get a house. There was an outside bigwig investor type that was willing to buy it all at a certain price. And I said, "Rather than sell it to somebody who's already got a lot of money, why don't I give the Apple employees the opportunity?" We were going to go public soon and it was going to be worth a lot more (and was eventually), so basically I sold it to 40 Apple employees. Our legal department was very concerned because they were supposed to be sophisticated investors. They finally gave me the OK. I did the deal and sold it to them, and they each pretty much got a house out of it.

**Livingston:** That was so generous.

**Wozniak:** But it's that whole thing I was talking about: Hewlett-Packard, we're a community. There was a recession in '73 and Hewlett-Packard had to cut back 10 percent. Instead of laying off 10 percent of the people, they cut everyone's salary by 10 percent and gave us one day off every two weeks. So basically they said "nobody goes without a job." And I like that sort of thing. So a bunch of Apple engineers and marketing people got to benefit from going public. Otherwise, they'd have no stock at all. Mike Markkula kind of felt that some of these people didn't deserve it; some people shouldn't get stock. But I disagreed with him on that. Nobody stopped me, so I did it.

**Livingston:** But you still kept enough stock for yourself to buy a house, right?

**Wozniak:** The money I got from Apple employees, I used to buy a house. It was kind of an early state to be selling out 15 percent of your stock, but hey, that was a great opportunity for me. When I designed the Apple stuff, I never thought in my life I would have enough money to fly to Hawaii or make a down payment on a house. So it was huge deal for me.

Steve Jobs (left) and Steve Wozniak (right) in 1975 with a blue box
Photo by Margret Wozniak