

John C. Dvorak

PIONEERING HARDWARE



Second of a series

After writing last week's column about successful hardware products, I talked to some **old timers**. Remember that an element of a successful hardware project is a memory-mapped video output. The first to popularize the concept was **Don Lancaster** with his TV typewriter stuff in the early '70's. Nearly all experimenters were greatly influenced by Lancaster. In many ways, Lancaster can be considered the **father of the personal computer**. Even Michael Swaine, the coauthor of *Fire in the Valley*, agrees with me on this.

Lancaster, who now has a number of **excellent** books out, including the two-volume *Don Lancaster's Micro Cookbook* (available from Howard W. Sams & Co) is an unsung microcomputer hero. When articles are written about "pioneers" or when trade shows have a "pioneer panel", seldom does Lancaster's name emerge.

Listen to the excerpt about Lancaster from *Fire in the Valley* in which he is quoted from an old article:

Lancaster, one of the more prolific contributors in electronic magazines was nothing less than a visionary...

"Obviously, it's (the 1973 TV Typewriter) a computer terminal for timesharing services, schools, and experimental uses. It's also a ham radio teletype terminal. Coupled to the right services (which didn't exist then), it can also display news, stock quotations, the time, and weather. It can be a teaching machine, particularly good for helping smaller preschoolers learn the alphabet and words."

While this wasn't a personal computer yet it was definitely the **progenitor** and made for smooth acceptance of the later-to arrive memory mapped SOL, VDM-1

terminals, and the Apple II.

Lancaster was to influence nearly all of those designers that came after him. Designers including **Stephen Wozniak**, who was not (according to him) in the computer hobbyist mainstream at the time he created the memory mapped scheme for the Apple I, were often not directly influenced by Lancaster. The marketplace and "insiders" were influenced. And the near **instant success** of the Apple I was partially prepared by Lancaster's influence to the "insiders" These are the guys who would say: "Yeah, that's the way to do it — it's great."

As an aside, it is interesting how Woz, as an independent designer, hit the **bull's-eye** with his sheer, inherent skill. At this point we have to wonder if we all would have been better off had the Apple II been only a minor success — so Wozniak would have to design for a living. I'm sure we all would have some **dynamite toys** to play with. As it stands, he's a guy who can't top his first act — and doesn't have to.

It is also interesting to note that Hewlett-Packard (where he worked before quitting in frustration) **wouldn't** let Wozniak do any work with computers or research and development "because he didn't have a college degree". If I were David Packard, I'd hunt down and **boil in oil** the managers who failed to identify Wozniak as a budding genius. After all, the company could have sent him to school if it were that **preoccupied** with sheepskin. I wonder what HP would do with Einstein if he were new to the staff. The firm would probably have him **wire wrapping**. You wonder about these things when you see things such as the HP-150, which the company thought would rule the world.

Back on track. The point is that Lancaster can be considered the father of the personal computer. What he did in 1973 isn't all that much different from what's happening today. Take the early automobile industry, for example. We remember the Walter Chryslers, Henry Fords, and even such late-comers as Ferry Porsche. Few remember Frank and Charles Duryea, the U.S. automobile inventors. The latter two **ended up on the sidelines** when it was all said and done. Sometimes when you're way out in front, people forget you're even in the race.

OK, so much for a tribute to Lancaster.

Let us get back to the requirements for hardware access.

Another element of the **perfect personal computer** is to include an easy-to use programming language. And this does not mean Pascal — **it means Basic**. There is no successful machine (and there never will be in our lifetimes) that cannot be programmed with Basic, the people's language. It's all part of the third-party support concept I raised last week.

Although Basic is **not** the world's greatest language, I submit that its ease of use makes it the perfect helpmate for the "guy with the idea" Which means that the nonprogrammer who has an idea for a program.

He writes the Basic program and shows (maybe sells) a completed version — that is essentially the prototype. A sharpie can come in later and recode it. **This is the key** to third party mass support.

Although professional programmers and **computer snobs** won't admit it, there are more good ideas among the 230 million laypersons than among all the professional programmers in the world. If the masses of enthusiastic laypersons (a subset of the general public) are not encouraged to support a computer — it's dead.

Now it would be nice if someone could find a language better than Basic, but at this point it may not be possible. Professional programmers who can theorize and code a language continue to **include the arcane** (such as a reverse-polish notation) or a limit (such as a forced structure). They all **miss the point** (or could not care less) about encouraging any additional amateurs to learn to program so we can get those one or two guys with a good idea to implement the good idea.

This will not happen if those one or two guys are forced to get their degree in computer science.

Because many of the computer scientists of the world are still thought of — and consequently do think of themselves as **second-class mathematicians**, they do spend much of their energy in a **lifelong pursuit** of self-worth. This takes the turn of turning the simple into the complex to prove they're smart. Heaven forbid that one of these guys should "waste" his time designing a programming language my mother could use. □