# MIPS COMPUTER SYSTEMS

*"In the 1960's, the invention of integrated circuits allowed the main frame computer to be built, and IBM dominated the field. In the 1970's, large scale integration enabled a new kind of computer, the mini, and a new player, Digital, emerged to dominate the new generation with its VAX line. Later, the development of the microprocessor again enabled a new generation of computers (PC's and workstations) and another new force, Apple, was born. RISC is an equally momentous technology. It will enable the computing generation of the l 990's: network computing. MIPS is fighting to be the dominant force behind this new generation. "*

*-- Co-Founder Skip Stritter*

*"It's as if the company was Luxembourg at the beginning of World War II except it knew how to make the bomb. RISC was thought of as the next ultimate weapon in the computer industry, but without a track record MIPS needed to win some friends so it could borrow a factory and an air force. "*

*-- CEO Robert Miller*

This case was prepared by Charles B. Krenz under the supervision of James C. Collins, Lecturer in Business, Stanford University Graduate School of Business. Revised by L.A. Snedeker in August of 1990, and by Nancy Katz in March of 1993. Production of this case was funded by John M. Goodfellow (MBA 1955) and Harvey R. Laderman (MBA 1958).

***The Situation***

In May of 1987, three years after its founding, MIPS Computer Systems was in turmoil. MIPS had only $700,000 of its original $20 million capitalization left in the bank. At the current burn rate, this amount would last less than a month. CEO Robert Miller, who had joined MIPS only a week before, was saddled with the task of leading MIPS out of these dark days and into a position of prominence in the emerging computer revolution -- a revolution based on RISC microprocessor technology coupled with the UNIX operating system.

***The Technology***

MIPS began as a laboratory project. John Hennessy, a professor of electrical engineering and computer science at Stanford, was fascinated by a new microprocessor design called "Reduced Instruction Set Computing" or "RISC." As its name implies, RISC simplified the number of instructions computers need to operate, and thereby significantly increased the speed at which data were processed over the then-current technology. Intrigued by the potential of this infant technology, Hennessy was joined by a visiting professor, John Moussouris, and Stanford computer science Ph.D. Skip Stritter on a project they called "Microprocessor without Interlocking Pipeline Stages," or "MIPS" for short. Not coincidentally, MIPS also stands for "Millions of Instructions Per Second," the scale used to measure the fastest computer systems. Like the Stanford projects from which Sun Microsystems and Silicon Graphics emerged, MIPS received research funding from DARPA, the Defense Department's Advanced Research Projects Agency. By 1983, the project had produced a prototype RISC microprocessor that ran at 2.5 MIPS, 10 times faster than the fastest IBM PC then on the market.

[For more details about microprocessors and RISC technology, please see the glossary on page 18.]

***Company Founding***

Armed with their prototype, the three founders set out to develop a commercially viable RISC product which they hoped would lead to a revolution in computing. MIPS' goal was to become a dominant industry player, much as IBM, Digital and Apple had done in previous generations. (See Exhibit 1.) However, unsure of how to best exploit this new technology, the three knew they would be competing against the biggest electronics firms in the country. IBM, Hewlett Packard and UC Berkeley were all working on RISC research projects, and other companies were rumored to be interested.

Clearly, MIPS needed significant capital to make this goal a reality. However, the Mayfield Fund, a leading Silicon Valley venture capital firm with strong ties to Stanford, believed in the new technology and acted as sole and founding investor when MIPS was incorporated in August of 1984. The firm received two seats on MIPS' board of directors, and coordinated the search for corporate officers. By early 1985, Vaemond Crane replaced an interim president to become MIPS' first permanent CEO. Less than a year later, the young company shipped the world's first RISC microprocessor. This unit, the R2000, operated at 5 MIPS.

Attracted by RISC's performance, the U.S. government remained interested in MIPS' technology. To further this interest, MIPS hired a salesman/lobbyist to spread the word in Congress and at the Pentagon so as to get MIPS microprocessors designed into as many new weapons systems as possible. Wherever compact, high-powered computing systems were needed, MIPS had a potential customer. For example, the Air Force purchased MIPS' microprocessors for development of complex radar and target recognition systems. MIPS was a finalist in the competition for a bid to develop control technology to be used in the Advanced Tactical Fighter. The Navy used MIPS microprocessors when it upgraded the electronics in its P3 Orion submarine hunting aircraft.

Although the government proved a logical entry point for RISC, MIPS' management believed that the more conventional, non-military computing systems presented the best opportunity for their new technology. Refinements to existing architectures had been slow and evolutionary, leaving industry experts to postulate that microprocessor manufacturers Intel and Motorola were "dragging their feet" in developing revolutionary performance advances.

While MIPS' processors clearly represented the revolutionary increase in speed that major users were seeking, they also carried high switching costs. Unlike industry leaders Intel and Motorola, who assured customers that their existing software applications would run unaltered on new versions of their microprocessors, MIPS had to prove that its speed advantages were significant enough to compensate customers for the costs of modifying their existing software.

*Price: Performance*

Fortunately, the speed advances were dramatic. Price:performance measures (dollars per m.i.p.s.) revealed that MIPS possessed at least a 10:1 advantage over existing mini-computer systems. In some cases, the ratio was 40:1 ! More impressive, MIPS' RISC architecture was evolving so quickly that its microprocessor speed was doubling every 18 months, compared to an industry average of three years.

However impressive these performance increases, the large-scale adoption of new hardware required that third-party software vendors (such as Lotus and Microsoft) develop compelling applications for the hardware that would attract customers to that particular "family" of computers. Software written for one particular microprocessor family, such as the Motorola 68000 series used in Apple's Macintosh computers, would run on that family exclusively. Similarly, IBM PCs and compatibles ran only on Intel's 8080 series. Because of the high fixed costs associated with developing software for a particular series, software publishers were reluctant to develop applications until the series' customer base had been proven.

*The UNIX Operating System*

Because of the need for applications to run on its systems, MIPS adopted the UNIX operating system as a standard for its products. The operating system translates applications (such as *Microsoft Excel* and *Lotus* 1-2-3) into the language the hardware understands. It enables the computer to print, read and create files, receive information from the keyboard and send information to the screen -- all of its most basic functions. By adopting UNIX, the industry standard operating system among the large organizations in MIPS' target market, the company decreased the switching costs associated with adopting its RISC systems and ensured that there would be a large body of applications software that would run on them.

The UNIX system had been evolving since Bell Labs had first released it in 1969. Its resilience reflected its widespread use on the university campuses where most computer scientists and programmers learned their trade. In the early 1980s, UNIX was growing in popularity, and had been adopted by computer manufacturers Pyramid, Plexus, Altos, HP and Sun for use on small systems.

In summary, to gain broad acceptance in the marketplace, a new class of computer systems had to achieve the "critical mass" necessary to attract software developers to its architecture. For the design to be successful, all of the pieces shown in Figure 1 needed to be in place. However, no more than one or two designs would likely emerge as industry standard RISC-based microprocessors. MIPS resolved that its microprocessor designs would be one of those few.

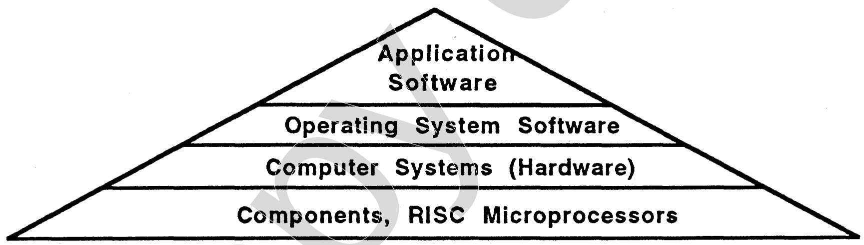



**Figure 1: Ingredients of a successful computer architecture**

*1985: A New Strategy*

By 1985, there appeared to be three different product categories which would have to be developed if the RISC revolution was to be successful. MIPS had the potential to participate in all three.

A) *RISC Microprocessors.* MIPS designed these chips, but could not afford to invest the capital necessary to become a "foundry" or microprocessor manufacturer. The economics of microprocessor fabrication favored large production runs at low margins, which would not be possible given the current demand for RISC chips.

B) *RISC Workstations.* Producers of RISC-based "super-workstations" would compete with existing workstation manufacturers. Workstations are highly powerful personal computers primarily used in scientific and engineering applications such as circuit design, stress analysis and computer-aided drafting. Faster than traditional PCs, workstations are better able to handle these complex applications, and can transfer information rapidly among themselves when connected via a network.

*C) Software.* MIPS could capitalize on the increased importance that software would play in the new architecture by creating applications for RISC-based workstations produced by other manufacturers.

[Appendix B presents the opinions of several industry experts on trends in the emerging workstation industry.]

Within MIPS, each alternative was championed by its own constituency, with the largest group favoring development of full systems based on MIPS microprocessors. This alternative would bring MIPS into direct competition with Sun Microsystems, the dominant workstation manufacturer, and Intel, the world's largest microprocessor manufacturer. However, to be a force in the computing industry, MIPS knew it had to do more than simply build chips.

Because of its larger vision, the company decided to take option B, to produce computer boards built around MIPS microprocessors. Traditional computer manufacturers could then build whole systems using these boards. In its initial marketing, MIPS would target computer manufacturers who lacked an installed base -- companies wishing to launch 'attack" products designed to grab market share by attracting high end users to their systems' speed.

Meanwhile, workstation manufacturers Sun and Apollo were interested in the potential of RISC systems. Both companies contacted MIPS in hopes of procuring microprocessors alone. However, intent on it strategy of selling only compete boards, MIPS refused. Since Sun and Apollo were interested only in MIPS' chips, neither pursued a deal with MIPS.

Further demonstrating its commitment to its board strategy, MIPS completed a prototype computer system, the M500, in December of 1985. MIPS designed the M500 for software publishers to use in developing programs for its RISC architecture. In addition, the M500 would highlight the potential speed of RISC workstations. However, the system's design was crude. According to Al

Sisto, MIPS' vice president of marketing, "its microprocessor was like a beautiful diamond in a terrible setting. The M500 was a mess; jumper wires made pin connections that hadn't been etched in its boards."

***Building a Solid Technical Team***

Despite the M500's shortcomings, the technical reputation of MIPS' design team gathered momentum as competing projects at other companies failed to produce workable designs. By mid 1986, MIPS' engineering department had swelled to 65 people, many of whom had defected from RISC projects at industry giants like HP, IBM, Data General, Intel and Motorola. MIPS' success reflected an environment specifically created to engender technical excellence. According to Jake Vigil, vice president of engineering and manufacturing, CEO Vaemond Crane deserved the credit for this accomplishment: "The dominant activity of a start-up in its early years is product development. Under his leadership, an environment had been established where we could create revolutionary products."

***Production Problems***

Although MIPS' design was clearly the fastest in its class, manufacturing this new generation of microprocessors was proving difficult. Since MIPS still did not have its own manufacturing capabilities, all production was subcontracted to outside manufacturers. This lack of control over delicate manufacturing processes was proving disastrous to MIPS' strategy.

Microprocessors are made by focusing miniaturized images of circuit designs onto wafers of silicon in a process similar to printing photographs. Under perfect conditions, each wafer would yield 81 microprocessors. Since each of these microprocessors consisted of thousands of transistors, MIPS' design required the flawless reproduction of 80,000 transistors in an area about the size of a thumb nail. In addition, while state-of-the-art technology could accurately reproduce images 2 microns in width, MIPS was asking for .8 microns.*

Because of these challenging specifications, initial production runs yielded not 81 chips per wafer, but rather only *one* chip for every *three* wafers. Over 99% of these early MIPS microprocessors were rejects!

* There are one million microns in a meter.

*Semiconductor Partner Strategy*

MIPS' Semiconductor Partner Strategy emerged from the company's needs for both better production capabilities and increased capital. The strategy emanated from an offer made to Thompson S.A., a European conglomerate, and was later expanded to other microprocessor manufacturers. The negotiations with Thompson, which went on for most of 1986 and continued on into 1987, illustrate the strategy.

For a fixed fee ($7 million in this case), Thompson would receive 10% ownership in MIPS and a license to manufacture MIPS' microprocessors. In exchange, MIPS would receive royalties on each microprocessor Thompson manufactured and sold, but would give up the rights to market RISC chips in Europe. Thompson was the largest manufacturer outside of the United States of microprocessors for military use, as well as a qualified second source for Motorola microprocessors. In addition, their manufacturing techniques were identical to those of Sierra Semiconductor, MIPS' most successful supplier. Because of Thompson's production capability and established distribution, MIPS' management viewed Thompson as an ideal Semiconductor Partner.

*1986: Abandoning the Board Strategy*

Having never generated even a quarterly profit, MIPS set an ambitious 1986 revenue goal of $8 million. This goal increased the company's focus on sales. While MIPS had planned to sell the M500 system for development purposes only, it began producing the system for several vendors to resell as a finished product. Still lacking its own manufacturing facilities, MIPS subcontracted all aspects of the M500's production except for the final quality control inspection.

Meanwhile, sales of boards proceeded below expectations throughout 1986. As a result, MIPS agreed to abandon its "boards only" rule late in the year and sell chips to Silicon Graphics Inc. (SGI), a promising young workstation company. SGI's workstations, designed to process sophisticated and complex graphics for engineering and design purposes, required the speed of a RISC-based chip, and could showcase the benefits of MIPS' design. By "betting the company" on the eventual success of MIPS' chips, SGI provided a much-needed vote of confidence for MIPS.

An entirely different opportunity presented itself when Prime Computer approached MIPS late in 1986. The company was interested in distributing a state-of-the-art workstation under the Prime label. Under the terms of the deal, MIPS' engineers needed to re-configure the M500 to Prime's specifications. While the engineers agreed that the reconfiguration was possible, they were adamant that the proposed delivery schedule was not. Although MIPS did not get the job, the episode both exemplified and exacerbated the company's increasing factionalization. The engineers' complaints had reached the Board: "Marketing is making promises the engineering department can't keep."

By the end of 1986, MIPS was finally operating in the black, fueled by impressive fourth quarter sales of $4.5 million. Despite this success, rumors continued to surface, such as: "It's not real sales, just a reduction in back orders;" "The stuff we're shipping probably isn't going to make spec;" and "We'll have to eat it -- all of it -- when it gets returned."

***1987: The Situation Worsens***

Tensions continued to rise in the company. The $7 million Thompson deal still had not been signed. "Every week at the all hands meeting we heard the same story: it should be signed next week." The relationship between engineering and marketing continued to worsen. In addition, CEO Vaemond Crane abruptly left MIPS to pursue other opportunities. The temporary CEO the Board hired was unable to improve the situation, as exemplified by the comments of a high-level member of the engineering staff: "The guy was a consultant who didn't know anything about high tech. He looked at the cash balance, saw where things were going and laid people off. It was as simple as that... I was beginning to wonder if the Board sent him in to shut the company down." Cash projections for the upcoming quarter revealed the situation's urgency. (See Exhibit 2 for the cash projections, and Exhibits 3 and 4 for the income statement and balance sheet, respectively.)

Al Sisto, MIPS' head of marketing and sales, summed up the situation as follows:

*"Those were tough times. I can remember driving over to Silicon Graphics to pick up a check so we could make payroll. We were trying to sell anything we could to make our customers happy. I think the engineers wanted to build workstations and be another Sun. They were off doing what they wanted to. We just didn't have enough middle managers."*

On April 27, 1987, MIPS' fourth CEO, Robert Miller, reported for work. (See Exhibit 5 for MIPS' organization chart.) A. Grant Heidrich, a general partner at Mayfield Venture Capital, describes the decision to hire Miller:

*Never did we lose faith in the basic technology. Bob was the only one we interviewed who shared our vision for the future: design and produce a RISC architecture capable of running the UNIX operating system. In addition, produce a new generation of system software and compilers which will make our machine's RISC design transparent to the applications programmer. Bob also had the required interpersonal skills and operational competence.*

An article on Robert Miller's motivations for coming to MIPS is included as Exhibit 6. Comments from Miller on his philosophies on business and life are contained in Exhibit 7.

***The View from the Top***

The list of problems facing Miller was long and complicated, ranging from short-term survival to long-term strategy. Although he was impressed with the company's technical capabilities, the marketing program was one of Miller's greatest concerns:

*In my model of the world, the technical team is the engine that runs the ship. I felt that I had a pretty good engine room. Using this analogy, I looked at the sales and marketing organization. They tend to be the navigators; they set the direction. I felt I had a pretty weak navigator. When you want to sell something you start by defining what sort of person wants to buy it, then how you're going to reach them and then how you are going to convince him that yours is better than the next guy's. There was none of that. They couldn't tell me who the ideal customer was.*

Although MIPS still had the most advanced technology, the competition was beginning to catch up. Product announcements and rumors of impending announcements from competitors made the "RISC war" even more turbulent. Hewlett Packard and IBM had both introduced RISC-based products. SUN Microsystems had announced a workstation based on its own proprietary RISC technology (SPARC). Although MIPS considered Sun its most formidable competitor, engineers at MIPS remained cautiously optimistic. Sun's first product had no software and did not integrate well into its existing product line. Furthermore, MIPS believed that Sun's entry legitimized RISC technology as a viable design concept in the eyes of many prospective customers. Even so, Miller felt that MIPS' grasp of market conditions was inadequate:

*I was absolutely convinced this company did not know where it was competitively or where it was as perceived by its customers. They didn't know if their customers either loved or hated 'em. (There was in fact a combination of those two emotions.) MIPS didn't know what leverage it had either on the customers or the competition and couldn't answer the question: where do you want to be two years from now?*

Miller faced another critical issue: MIPS' deteriorating revenue and cash situation. The revenue forecast for 1987 was $24 million, but Miller expected MIPS to fall far short of this target -- perhaps to less than half. While MIPS had $700,000 in the bank, payroll for the 123 employees was over $600,000 per month, and rent on its 80,000 square feet facility was $200,000 per month.

### *Surveying the Options*

One possibility for conserving cash was to lay off 30% to 40% of MIPS' employees, keeping the engineering team intact and cutting everything else. Given the already strained atmosphere in the company, Miller had to consider this decision especially carefully. As he described the situation: "There was incredible cynicism and frustration. People were not about to trust in anything. There was a tremendous concern that the company was withering on the brink. Attrition was very high. There was just a tremendous lack of confidence."

Another option was to eliminate a costly engineering project whose purpose was to perfect a circuit design known as Emitter Coupled Logic. This project, which provided evolutionary improvement to the physics of circuit implementation, was not crucial to the RISC design. While ECL promised to quadruple microprocessor speed, the engineers couldn't be sure when the project would be completed; they hoped for two years. Since the ECL project was costing $4 million per year, eliminating it would save MIPS a great deal of money. However, some felt that it might result in MIPS' losing its long-term technological edge.

Production problems continued to plague MIPS. Only Sierra Semiconductor had proven to be a reliable supplier of MIPS designed microprocessors. Efforts to qualify a second source in Japan had not been successful.

The Semiconductor Partner Strategy posed a difficult dilemma for MIPS. The Thompson deal was still not signed. If completed, it would provide MIPS with much needed cash and the second manufacturing source it required. However, in exchange MIPS would give up 10% of the company for $7 million and all rights to sell directly in Europe.

If the Thompson deal did not work out, the question remained as to whether MIPS should pursue this strategy with other partners. Proponents of the strategy cited the advantages of the cash infusion and production capabilities it would provide, as the prospects for MIPS' manufacturing its own chips became increasingly dim due to the high costs involved. They also cited the potential access to distribution channels that partners might bring. Miller observed:

*We did not have a distribution channel. There are those who think that the world begins and ends with the product and that the distribution channel will somehow materialize and evolve. That's absurd. The idea that MIPS was a systems company without any means of distribution or ability to support it was bizarre.*

Opponents of the Semiconductor Partner Strategy argued that it was risky because it required the transfer of highly proprietary design information to potential competitors with far greater resources. Furthermore, Miller had held preliminary discussions with other potential partners, such as Motorola, Intel and NEC, and none had been interested. The only companies that expressed interest were second-tier players that did not have the market power or distribution channel to "make the market."

Another possibility was to consider an outright merger with a larger company or one of MIPS' significant customers.

One last alternative was to persuade one of the major computer makers -- such as IBM, DEC or HP -- to design MIPS' RISC microprocessors into some of their product lines. If successful, this strategy would give the small company a gigantic jump in credibility and very large order volumes. However, no one at MIPS had made any focused efforts in this direction.

The question of determining MIPS' strategy remained. Should it simply sell chips? Should it design, build and sell entire workstations in an attempt to be the first company to put true mainframe power in a desktop computer? Should it develop software for the RISC-based computers sold by others'? How exactly should MIPS take advantage of its technology'?

In reflecting on these question, Miller shared some of his own thoughts:

*I came to MIPS because of the core technology and the market potential of that technology linked with the UNIX operating system. Forecasts of the market for computers using the UNIX operating system are in the billions of dollars, and I believe that RISC microprocessors will be at the heart of it all.*

*I'm a great believer that the way you succeed in this industry is by owning your technology. We own a technology called the MIPS RISC microprocessor, but few people seem to understand the significance of that. A complete technology is the silicon and the software -- and we have both. Other companies only have the silicon. At Data General, when I wanted to buy a Motorola microprocessor, I not only had to buy the chip from Motorola, I had to negotiate with ten different software companies plus do a lot of other work on the operating system to make the whole computer play.*

Linked with the question of what MIPS should try to do was the question of precisely *how* should Miller go about setting the direction for MIPS. Exactly what steps should he take -- what process should he employ -- to set the overall direction of the company? Fighting for its survival in this time of crisis, how much time should he dedicate to long-term thinking about what MIPS should strive to become?

## Exhibits

1. Glossary of Technical Terms
2. Computer Industry Growth By Segment
3. Cash Projections: March - May, 1987
4. Quarterly Income Statements
5. Balance Sheet
6. Organization Chart (May, 1989)
7. "Why DG's Miller Went To MIPS Computer," Electronics, April 30, 1987.

## Appendices

A. "RISC: Is It A Good Idea or Just Another Hype," *Electronics* , May 5, 1986.

B. "What's Hot and What's Not in Workstations," *Electronic Business*, March 15, 1987.

## Exhibit 1

## Glossary of Technical Terms

*This glossary is included to explain some of the technology related to the MIPS case, but is not knowledge necessary for discussion of the business situation.*

***Microprocessors*** Microprocessors are thin squares made out of silicon (about 1 sq. inch) that are the most basic element of a computer. Essentially, they are the "brains" of computers, responsible for executing tasks such as printing, storing or multiplying. They do this by successively breaking these tasks (or instructions) into smaller sub-instructions which are implemented throug h an array of electrical "on/off" switches. They do this with the help of software programs (composed of lines of code) that direct the information.

***Computer Boards*** A board represents a collection of electronic devices soldered on to a thin sheet of fiberglass, which are interconnected by electrical "pathways" etched into the fiberglass. Several boards, each performing a unique function, are connected to form a computer system (which can also include a a disk drive, a screen, or a keyboard).

***RISC vs. CISC*** The lines of software code that a microprocessor interprets are called the "instruction set." One theory was that the more detailed (and therefore longer) the instruction set, the more capable the microprocessor was of quickly getting the computer to do what you want it to do. This theory was the basis of CISC (Complex Instruction Set Computing)--which was used in the Macintosh, the IBM PC, and Digital Equipment Corp.'s VAX. Proponents of CISC's antithesis, RISC, argued that limiting the size of the instruction set enabled the microprocessor to more efficiently execute the vast majority of instructions by eliminating the "overhead" required to support the larger number of instructions. Comparing CISC to RISC is like comparing a 1000 page dictionary to a 10 page vocabulary list: RISC is faster to work with but not as comprehensive. A more complete discussion of RISC is included in Appendix A.

***Software***

Software basically consists of lines of "code" that a programmer inputs to get the computer hardware to do a certain function (word processing or spreadsheets, for example). It allows you to work with the keyboard and/or the mouse to process your information. Usually software is developed for a "family" of computers, and correspondingly, microprocessors. For instance, Macintosh software will run on *all* Macintosh computers, because they are all based on Motorola's family of chips. This compatibility is important to software developers because it expands the market for their software. Software developers want to be sure that a system will have a large enough market to justify development costs before they invest time and resources into a software effort. Often, though, customers will not want to purchase a system unless useful software is available. So developing a new system often has an inherent Catch-22.

***UNIX***

Every computer has an *operating system* that comprises lines of code that tell the computer hardware (e.g., the microprocessor) how to run. It makes it possible, for example, to receive information from the keyboard and direct it to the microprocessor so it does what you expect it to do. The operating system functions as sort of a mediator between the hardware and you. It also works with any software program (such as Microsoft Excel) that you are running to make sure it functions as you expect it to. A well-known operating systems is MS-DOS on IBM PCs (and compatibles). Many workstations use the *UNIX* operating system, including Sun, HP, NeXT, and Silicon Graphics.

## Exhibit 2

## Share of Computer Industry Growth By Segment

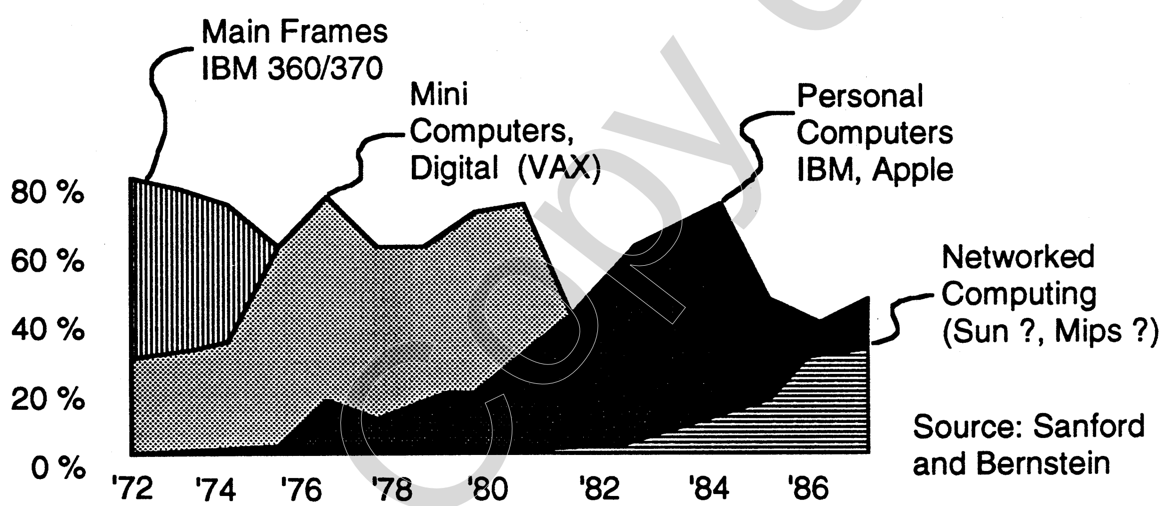



**Figure 1, Share of Computer Industry Growth By Segment**

# Exhibit 3
## MIPS Computer Systems
### March - May 1987 Cash Collection Schedule (Disguised Figures)

| Collections | Actual 15-Mar | Actual 22-Mar | Actual 29-Mar | Est. 5-Apr | Est. 12-Apr | Est. 19-Apr | Est. 26-Apr | Est. 3-May | Est. 10-May | Est. 17-May | Est. Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Company 1 | | | | | | 2 | 35 | | | | 37 |
| Company 2 | | | | | | 4 | | | | | 4 |
| Company 3 | | | | | | 2 | | | | | 2 |
| Tandem | | | | 242 | | 147 | | | | | 389 |
| Company 5 | | | 55 | | | | | | | | 55 |
| Company 6 | | | | | | 6 | | | | | 6 |
| Company 7 | | | | | | | 20 | | | | 20 |
| Company 8 | | | | 49 | | | | | | | 49 |
| Company 9 | 4 | | | | | 108 | | | | | 112 |
| Company 10 | | | 53 | | | | | | | | 53 |
| Company 11 | | | | 1 | | 2 | | | | | 3 |
| Silicon Graphics | 139 | | | 116 | 39 | | 207 | | | | 501 |
| Comps. 13-15 | | | | | | 5 | 77 | | | | 82 |
| Company 16 | 90 | | | 139 | | | 139 | | | | 368 |
| Comps. 17-21 | 0 | 0 | 15 | 92 | 173 | 0 | 45 | 39 | 0 | 0 | 364 |
| Company 22 | | | | | | 221 | | | | | 221 |
| Company 23 | 69 | | | | | | | | | | 69 |
| Company 24 | 72 | | 29 | | | | | 17 | | | 118 |
| Company 25 | | | | | 64 | | | | | | 64 |
| Company 26 | 145 | | | | | | 24 | | | | 169 |
| US Govt | | | | | | | 77 | | | | 77 |
| Company 28 | | | | 56 | | | | | | | 56 |
| Company 29 | 12 | | | | | | 6 | | | | 18 |
| Company 30 | | 4 | | | | | | | | | 4 |
| March Sales | | | | | | | | 100 | 100 | 200 | 400 |
| **Cash In** | 531 | 4 | 152 | 695 | 276 | 497 | 630 | 156 | 100 | 200 | 3,241 |
| | | | | | | | | | | | |
| **Cash Disbursements** | | | | | | | | | | | |
| Payroll | 328 | | 275 | | 333 | | 325 | | 333 | | 1,594 |
| A/P | 434 | 324 | 376 | 318 | 233 | 250 | 257 | 250 | 371 | 365 | 3,178 |
| Other: Westech | | | | 37 | | | | 37 | | | 74 |
| CGS | 113 | | | | 113 | | | | 113 | | 339 |
| Bank | 33 | | | 21 | | | | 21 | | | 75 |
| Europe | 80 | | | | 30 | | | 30 | | | 140 |
| AWARE | | | | 30 | | | | | | | 30 |
| COB | | | | 5 | | 5 | | 5 | | 5 | 20 |
| Rent, Bldg 1 | 36 | | | 36 | | | | 36 | | | 108 |
| Rent, Bldg 2 | 42 | | | 64 | | | | 64 | | | 170 |
| Contingency | | | | | 25 | 25 | 25 | 25 | 25 | 25 | 150 |
| **Cash Out** | 1,066 | 324 | 651 | 511 | 734 | 280 | 607 | 468 | 842 | 395 | 5,878 |
| **VC Cash** | 1,205 | 795 | | | | | | | | | |
| **Cash Balance** | 1,592 | 2,067 | 1,568 | 1,752 | 1,294 | 1,511 | 1,534 | 1,222 | 480 | 285 | |

Assumptions:
A) Payroll - Focal increases will be effective July 1, 1987
B) Accounts Payable is paid according to payment program with vendors
C) Accounts Receivable - Agressive collections, but attainable

# Exhibit 4

## MIPS Computer Systems
## Income Statement

(Disguised Figures)

| | 1985 year | 1986 1st Qtr | 1986 2nd Qtr | 1986 3rd Qtr | 1986 4th Qtr | 1986 Year | 1987 1st Qtr |
|---|---|---|---|---|---|---|---|
| **Revenue** | | | | | | | |
| Hardware | | | $241 | $732 | $4,092 | $5,065 | $2,176 |
| Software | | $450 | $351 | $315 | $920 | $2,036 | $473 |
| Net Sales | | $450 | $592 | $1,047 | $5,012 | $7,101 | $2,649 |
| **Cost of Sales** | | | | | | | |
| Manufacturing | | $141 | $245 | $382 | $482 | $1,250 | $605 |
| Overhead Absorption | | | ($59) | ($40) | ($296) | ($394) | ($140) |
| Product cost | | | $100 | $329 | $1,863 | $2,291 | $1,024 |
| Other Cost | | $3 | $17 | $286 | $80 | $386 | $465 |
| Cost of Sales | | $144 | $303 | $957 | $2,129 | $3,533 | $1,955 |
| Gross Margin | | $306 | $289 | $90 | $2,883 | $3,568 | $694 |
| **Operating Expenses** | | | | | | | |
| Engineering | $3,233 | $1,412 | $1,570 | $2,394 | $1,397 | $6,773 | $1,904 |
| Marketing/Sales | $637 | $296 | $356 | $526 | $547 | $1,724 | $808 |
| Administration | $1,778 | $611 | $660 | $1,306 | $850 | $3,426 | $1,522 |
| Total Operating Expenses | $5,648 | $2,319 | $2,586 | $4,226 | $2,794 | $11,923 | $4,234 |
| Operating Profit | ($5,648) | ($2,013) | ($2,297) | ($4,136) | $89 | ($8,356) | ($3,540) |
| Other (Income)/Expenses | ($260) | ($104) | ($185) | ($106) | ($5) | ($401) | $68 |
| **Profit Before Taxes** | ($5,387) | ($1,910) | ($2,111) | ($4,029) | $94 | ($7,955) | ($3,607) |
| **Net Profit After Taxes** | ($5,387) | ($1,910) | ($2,111) | ($4,029) | $94 | ($7,955) | ($3,607) |

# Exhibit 5

## MIPS Computer Systems

## Balance Sheet

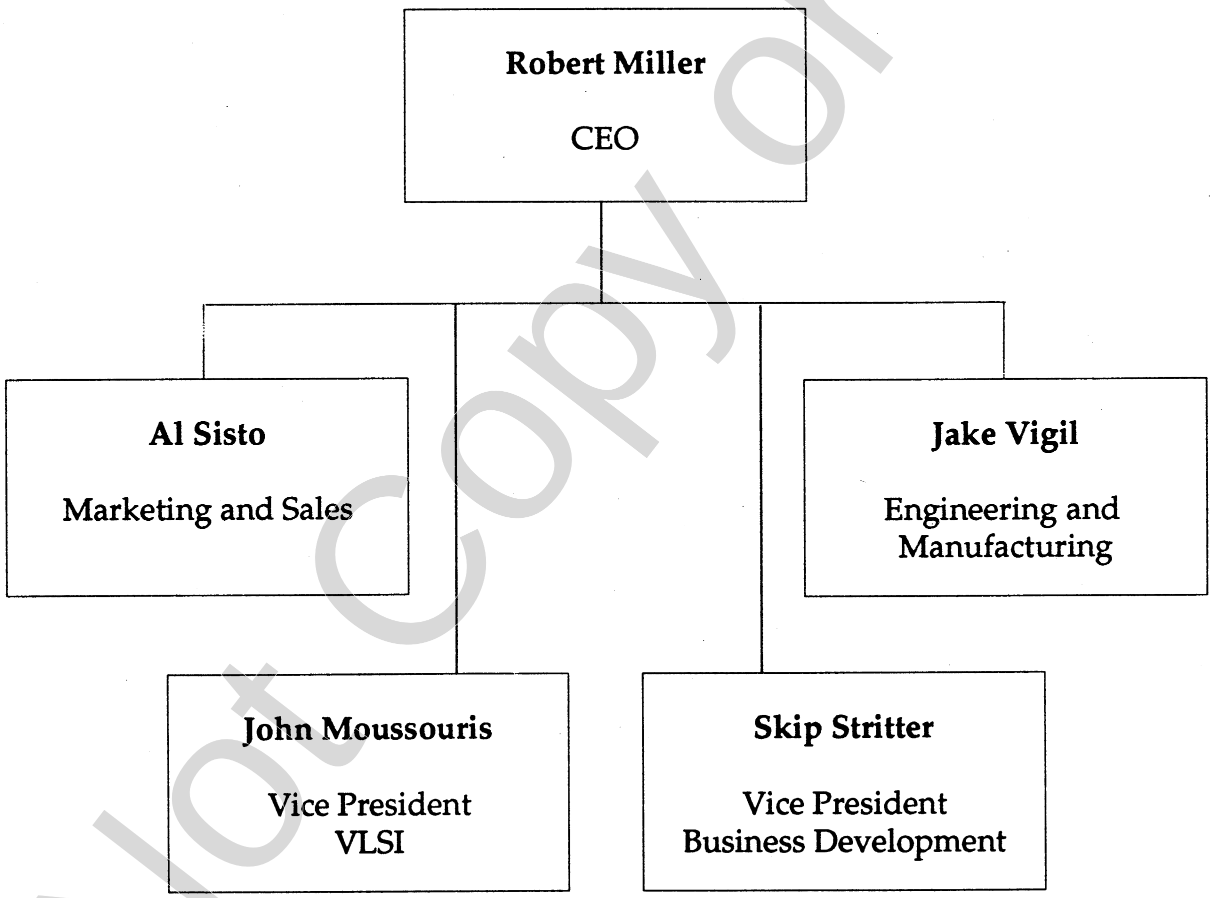
(Disguised Figures)

| | Dec. 31, 1986 | Apr. 30, 1987 |
|---|---|---|
| **Current assets:** | | |
| Cash and Short term investments | $2,988 | $777 |
| Accounts Receivable, net | $5,210 | $3,326 |
| Inventories | $1,649 | $2,422 |
| Other current assets | $228 | $205 |
| **Total current assets** | **$10,075** | **$6,730** |
| **Property Plant and Equipment** | | |
| Machinery and Equipment | $6,300 | $6,600 |
| Furniture and fixtures | $1,806 | |
| Other Assets | | $1,121 |
| Leasehold improvements | $121 | |
| Accumulated Depreciation | $1,110 | $1,805 |
| **Net property and equipment** | **$7,117** | **$5,916** |
| Restircted cash | $1,100 | $1,105 |
| Deposites and other assets | $221 | |
| **Total Assets** | **$18,513** | **$13,751** |
| **Liabilities and Shareholders Equity** | | |
| **Current liabilities:** | | |
| Accounts Payable | $6,009 | $3,451 |
| Current portion of LTD | $181 | $1,609 |
| Other Current Liabilities | $2,328 | |
| **Total Current Liabilities** | **$8,518** | **$5,060** |
| Deferred revenue | $984 | $1,285 |
| Long term debt | $363 | $2,282 |
| Other Long Term Liabilities | $1,844 | |
| **Total Liabilities** | **$11,709** | **$8,627** |
| **Shareholders equity** | | |
| Preferred Stock | $20,336 | $20,336 |
| Common stock | $377 | $666 |
| Subordinated Debt | | $1,800 |
| Accumulated Deficit | ($13,909) | ($17,678) |
| **Total Shareholders Equity** | **$6,804** | **$5,125** |
| **Liabilities and Shareholders Equity** | **$18,513** | **$13,752** |

## Exhibit 6

## MIPS Computer Systems
## Organization Chart

## May, 1989



PEOPLE

# WHY DG's MILLER WENT TO MIPS COMPUTER

BOSTON

A senior vice president at Data General Corp. who also has 15 years of experience at IBM Corp. under his belt is the kind of quarry that computer-industry headhunters love to bag. Little wonder, then, that Robert C. Miller, who fits that description, had several "career opportunities" dangled before him during his six years at the Westboro, Mass., computer maker. But only one seemed worth taking—the chance to become president, chief executive officer, and chairman of MIPS Computer Systems Inc. of Sunnyvale, Calif.

"It's always been a dream of mine to run my own company," says the 43-year-old Miller. He headed Data General's Information Systems Group, which accounted for revenues of $1.2 billion in 1986, until he signed on with MIPS in mid-April. "To find one that played to my interests like MIPS is truly a rare opportunity."

MIPS, a private company formed in 1984, was the first firm to specialize in reduced-instruction-set-computer technology [*Electronics*, April 29, 1985, p. 36]. It remains as perhaps the only company dedicated to a pure RISC approach. Its product line includes commercial versions of a RISC chip originally developed at Stanford University, plus board-level products, a component kit, optimizing compilers, and a development system built around that chip.

The RISC technology was one reason Miller accepted the MIPS offer. The company's location, in Palo Alto, Calif., also played a role, as did the opportunity to head "a team that has real class." He also likes MIPS' use of Unix.

**'DONE RIGHT.'** Miller earned a master's degree in thermodynamics from Stanford, which is in Palo Alto. "When I left there 20 years ago, I promised myself I'd go back. And when the people at MIPS took me through the architecture of the RISC processor, I could see that these fellows had done it right. They've moved the state of the art quite a bit forward," he contends. "These fellows" include John Hennessey, who pioneered in RISC when he designed the experimental chip at Stanford that MIPS is now selling. He remains at MIPS as chief scientist and now heads a Stanford research project aimed at developing an advanced version of the chip for the Defense Advanced Research Projects Agency.

Miller declines to discuss that effort until he's more familiar with it, but he's quite vocal about Unix and RISC. He's

**MILLER.** MIPS gave him an opportunity he says he simply couldn't overlook

convinced that RISC processors will be the dominant Unix platform for the 1990s. "Forecasts for the Unix market in the next decade are in the billions of dollars, and I believe this processor [and Unix] will be to the '90s what PC DOS is to the '80s," Miller predicts.

"I was convinced a few years ago that Unix was going to be an exciting part of the business," he continues. "To think about a processor board that runs at 8 mips at no cost premium versus a 68020-based board is especially exciting. We'll need to keep driving the performance up to realize high-function work stations and servers."

For now, Miller sees MIPS' current CMOS as the most cost-effective implementation for RISC architectures. But he anticipates opportunities for emitter-coupled logic in applications where users will pay a premium for speed.

Miller is joining a company that acting chairman Donald Muller says is aimed at becoming the dominant supplier of RISC-based products. MIPS has backing from some of Silicon Valley's top venture firms, including the Mayfield Fund and Kleiner Perkins Caufield and Byers, San Francisco. The company has already sold Unix-based RISC platforms to several original-equipment manufacturers, including Prime Computer, Silicon Graphics Computer Systems, and Racal-Redac. All in all, Miller concludes, "I couldn't overlook this opportunity."

*–Laurence Curran*

## Exhibit 8

*In the last question of an interview conducted in the preparation of this case, Robert Miller was asked to comment on his personal philosophy of doing business. His remarks are reproduced below.*

I'm a great believer in the concept of role models. I'm not a person who believes that everything should be invented from the ground up. If you can find some role models of people and companies that exhibit the kinds of traits and characteristics that you admire, the ideal thing is to synergistically put some combination of those things together. To put it in a familiar context: If you thought that Paul Rizzo at IBM was the best businessman you ever met, if you thought that Winston Churchill was the greatest charismatic leader in terms of the integrity of the leadership and the vision and the courage and all the things that went with it, and if you also believed that there were some people in your personal life who exhibited exceptional qualities, you should feel free to emulate these qualities. Parallel to that, if you thought there were things that IBM did very well in terms of how it treated its people, if you thought that there were things that Hewlett-Packard did well in terms of creating a good positive environment, if you thought that there were things that Tandem did well in terms of employee compensation, then put together a company that has all those things rather than doing it like a lot of people in the valley try to. They try to do it like no one has ever done it before.

The other philosophy I have is the toughest one to keep: Once you make a commitment you keep it. I think of business as a mission. People go on the mission and they don't always like the way it comes out. A lot of things get thrown in your path that you never predicted, but you should never lose sight of the things that you committed to do. Sometimes you can't get there in the time you promised or you can't get there within the level of spending because of things you really never could have expected. You never stop trying to make the commitment unless the other person releases you from it. You don't unilaterally drop the commitment. At all levels in this company, when we give our commitment we keep it. It doesn't have to be written down in a legal document. We've had a couple of situations where we've lost a great deal of money on verbal commitments made by salespeople who didn't have the authority to make such agreements, but we've kept the commitments. As a personal and business philosophy I see this in the Japanese. It's one of the things I admire most.

# INSIDE TECHNOLOGY

# RISC: IS IT A GOOD IDEA OR JUST ANOTHER HYPE?

## DESPITE THE FLAP, IT'S BEGINNING TO CHANGE COMPUTER DESIGN

*by Clifford Barney and Tom Manuel*

There's so much noise being generated now about the reduced-instruction-set computer and whether such designs are a big step forward that it tends to obscure what's really happening in computer architecture. The RISC concept is gaining increasing credibility as it starts showing up in major commercial designs. New products from Hewlett-Packard Co. and IBM Corp., for example, are helping a lot to move RISC out of the realm of debate and speculation and into the real world.

Yet the debate still rages on whether the RISC concept that was born a decade ago in the rarefied atmosphere of academia can be successfully applied to the design of commercial machines. Complicating things further is that computer scientists cannot even agree on just exactly what a RISC machine is.

The RISC concept emerged in 1975 at IBM's Thomas J. Watson Research Center in Yorktown Heights, N.Y. (see "How it all began," p. 29), when John Cocke, an IBM Fellow, came up with the idea. Cocke is considered the father of RISC, although the term was first used at the University of California at Berkeley, where graduate students under David A. Patterson designed a microprocessor they called RISC I. Other early RISC research was done at Stanford University in Palo Alto. As interest in the concept mounted, commercial RISC or RISC-like computers were produced by Acorn Computers, Celerity, Harris Computers, Pyramid Computer, Ridge Computers, Shiva Multisystems, and France's Thomson. Most recently, Mips Computer Systems, Sunnyvale, Calif., has introduced a blazingly fast "pure RISC" system based on the chip developed by John Hennessey at Stanford (see story, p. 56).

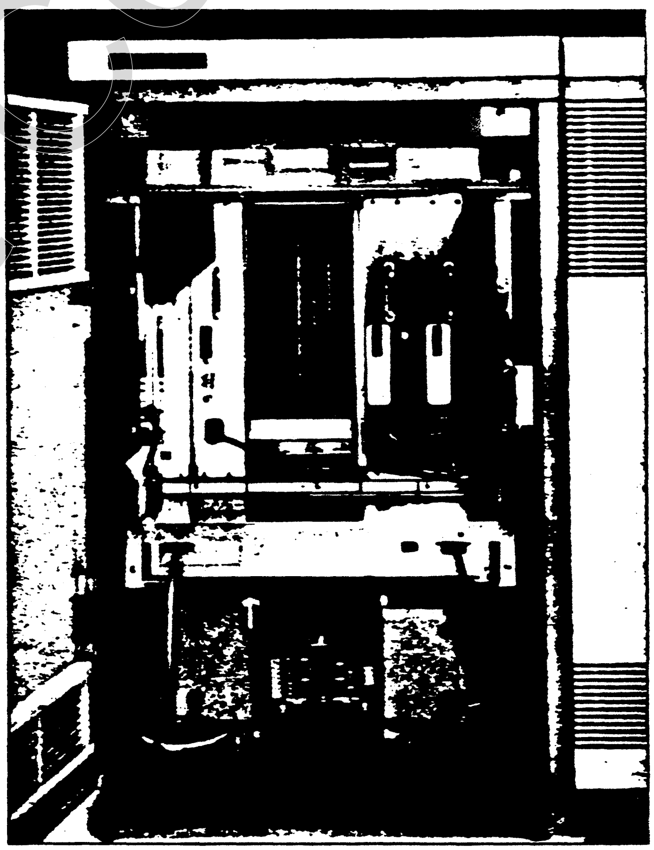
Now IBM and HP have minted two of the newest designs applying basic RISC principles. One is the IBM RT Personal Computer (see story, p. 34), and the other is HP's Precision Architecture [*Electronics*, March 3, 1986, p. 39], which is being used first in the new HP 3000 series 930 minicomputer. They both apply such principles as a small set of simple, regular instructions that execute in one cycle, only load and store instructions can access memory, other instructions operate only upon registers, hard-wired control is used instead of microcode, and complex compilers provide complex functions and generate the optimal code for the machine.

As interest has grown in the RISC concept, so has debate over it. The arguments swirl around the closely related questions of applying the university-nurtured technology to commercial machines and how to define RISC. In fact, the acronym has turned into such a buzzword that an argument over RISC versus complex-instruction-set computers practically drained all other sessions at Compcon in San Francisco in March.

Looking back on that debate—a panel discussion entitled "The Great RISC vs. CISC Debate"—G. Glenn Henry distinguishes between work done on RISC at universities and in industry. "The RISC chips done at UCB and Stanford really are the classical RISCs," says the IBM Fellow, who is also manager of hardware and software system development for the IBM RT PC at the company's Engineering Systems Products independent business unit in Austin, Texas [*Electronics*, April 28, 1986, p. 54]. "But RISC is a misnomer for the work being done in industry—for example, here at IBM in Austin and at HP.

"The commercial systems have some complex instructions where they are needed," he continues. "The classical RISC chips out of the universities are very simple—just chips, not complete systems. Therefore, it is rather unrealistic to compare performances of these to the performances of commercial systems."

Performance measurements cited for the RT PC were made using real applications running under AT&T Bell Laboratories' Unix operating system with virtual memory and error correction turned on and fully operating. They were not done with simple benchmarks on a stripped-down microprocessor, Henry notes. "But wait," he quickly adds, "I'm not putting the universities' work down. They did good work. They

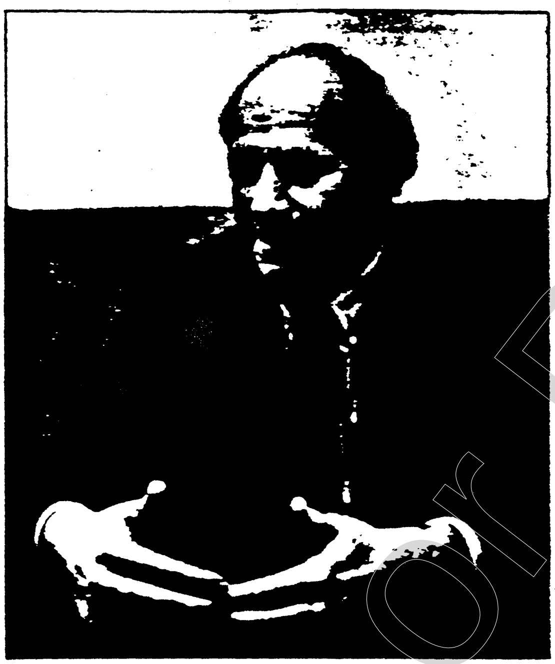


**COMMERCIAL RISC.** The HP 3000 series 930 is the first product from Hewlett-Packard built with its new RISC-based architecture.

learned things that we all could put to use."

Another IBMer rejects that notion. Nick Tredennick, a researcher at IBM's Watson Research Center, led an attack on RISC at Compcon. "RISC is a poor idea for commercial microprocessors," he said. "Commercial microprocessors are bandwidth-limited at the pins already, and reducing the instruction set makes the problem worse."

One argument against RISC is certainly that with fewer instructions, doing a particular task takes more instruction execution, concede Henry and Joel Birnbaum, an HP vice president and head of the Palo Alto company's RISC-like Spectrum project. That would mean that more memory bandwidth would be needed to deliver those instructions to the processor. "But," says Henry, "Tredennick works on putting a 370 on a chip, and for that task he can't use the classical university RISC. That's probably what he is talking about when he says that RISC is a poor idea for commercial microprocessors." Henry has proved that a commercial computer system can be designed using RISC principles while still solving the memory bandwidth problem.



**A BETTER MOUSETRAP.** HP's Joel Birnbaum believes RISC principles, rigidly applied, lead to better computers.

HP's Birnbaum says that "Tredennick is right, in one sense, to say that RISC won't work in a commercial microprocessor. But what's RISC? If you are talking about the Berkeley chip, as opposed to a set of design principles upon which you can build, that's such a trivial and obvious argument it isn't worth making. It's like arguing about how many angels can dance on the head of a pin."

David Patterson, the Berkeley computer scientist who helped start the debate four years ago, agrees that by now the argument is irrelevant. "In 1981 and 1982, it was heretical to suggest that you could improve performance by transferring hardware functions to software," Patterson says. "It was supposed to go the other way. In 19— rather than argue, let's see what people have built. The professional designers and the computer buyers will be the judges."

Even these judges, Birnbaum contends, won't be intent on classifying computers as RISC or non-RISC machines. "I don't know whether Spectrum is an architecture that proves that RISC is viable or that it is not," Birnbaum says. "The reason is that there is no 'RISC architecture.' There is a set of principles which can be applied in a particular architecture to solve a design goal."

These principles—such as single-cycle execution and absence of microcode—cannot be applied blindly to a commercial computer, Birnbaum adds. "We have some multicycle instruc-

## HOW IT ALL BEGAN

**It all began in 1975** when IBM Fellow John Cocke had an idea. Cocke, then a researcher at IBM's Thomas J. Watson Research Center in Yorktown Heights, N.Y., decided that a computer with a simple architecture, using a simple instruction set, could have certain advantages over the trend toward computers with increasing architectures and more complex instruction sets. In the late 1970s, graduate students at the University of California at Berkeley gave it the label it goes by today—reduced-instruction-set computer (RISC).

Cocke and his small team of researchers at first tested the idea's feasibility with a research prototype—the 801 computer (named after the number of the building where the team worked). The original project, however, did not have a new computer architecture as its objective. "We were studying what would be needed to make very large telephone switching systems, and a very fast controller was one of these needs," says Cocke, a 30-year IBM veteran. "After the switching project was abandoned, we started to consider the controller as a machine in itself." The result of the redirected effort was the 801 minicomputer prototype, which was built in 1979.

The research team experimented along many avenues to reach their objective of an efficient new architecture. It was not just slashing away at the instruction set. "In fact," says Cocke, "while it's true that we reduced the number of instructions, that is more a result than a cause. We didn't believe we should put complicated instructions into a machine when they can be built up from simpler ones without sacrificing performance."

**JOHN COCKE:** The inventor of RISC.

Cocke and his researchers also took advantage of advances in compiler technology. The new compilers were getting smarter at automatic programming and code optimization. The compiler technology made simple machine instructions feasible because if a complex instruction were needed, the compiler could string together simple instructions.

The 801 designers also put to use large, very fast memories. "Since the memory hierarchy of the 801 minimized problems of storing and retrieving information and we had designed the instructions to all be the same size and execute in one cycle," Cocke says, "we were able to build in a lot of overlapping—pipelining—of the execution of instructions." The 801 and other RISC machines pipeline nicely because of their simple organization and instructions. *–Tom Manuel* tions. So if you want to say we are not RISC, that's OK. On the other hand, most of the Spectrum instructions execute in one cycle. We have some microcode, to help with input/output. So we have RISC architecture, but it goes far beyond RISC."

Yale Patt, the Berkeley professor who chaired the Compcon session, calls RISC a "useless label," because "people can't agree on what it means." He suggests a new microprocessor category, "RISC-but," into which HP's Spectrum fits neatly.

In Glenn Henry's idea of RISC, "the key is to get the processor speed as fast as you can on the smallest amount of silicon and, at the same time, separate the processor from the bottlenecks of memory and I/O." This is necessary, he says, because in recent years very large-scale integration has produced processors that are much faster than memory and I/O circuitry. The goal is to make a single processor chip run as fast as possible, independent of memory and I/O speeds.

Henry says a computer architecture must incorporate three design concepts in order to be a true RISC. First, it must overlap processing with memory accesses. Second, instructions must operate in a single cycle. This generates the swiftest performance possible for a given VLSI speed; when the VLSI implementation can be scaled down even further, the processor will run faster. For example, if IBM builds its RISC microprocessor in 1.5-$\mu$m CMOS instead of the current n-MOS process, the chip will speed up significantly.

## EFFICIENT TARGETS

The third, and most subjective, qualification for a true RISC architecture by Henry's definition is that all instructions should be designed to be efficient targets for high-level languages. High-level-language compilers are then built to take advantage of the benefits of such architectural principles as the separation of storage from the processor. For example, the compilers must be able to organize instructions and data flow to use the architecture efficiently. "Therefore, we will get good compilers [as a fallout] from the work on RISC architectures," Henry says.

Henry's idea for the best term to replace the often misleading and confusing acronym RISC, is optimized-instruction-set computer (but OISC is more difficult to pronounce). "The optimized instruction set is optimized for compilers, optimized for maximum performance from individual processors, and optimized for not wasting silicon area," he says.

The opposite of optimizing the use of the silicon area is the temptation to exploit silicon technology. HP's Birnbaum considers complex instruction sets the outgrowth of what he calls "creeping elegance," which arises from this temptation. "There is an argument," he says, "that silicon is free—that we have reached a level of lithography such that it doesn't cost much to put extra transistors on a chip, within certain pin and power limits. So why be dumb enough to have only 100 instructions when for no extra hardware cost you can have 200? This is 100% fallacious. You pay for those extra instructions every time you execute any instruction, in several different ways."

One way is in design complexity, Birnbaum says. "If you have many instructions, you have many more conditions to

**ADVOCATE.** David A. Patterson of UC Berkeley was the first promoter of RISC principles for computer design.

prepare for and recover from. The design takes longer, and it is harder to change. You even pay in extended time to market." Patterson agrees, saying that with computer power increasing 20% to 40% each year, a complex design may lag the state of the art by the time it reaches the market.

Patterson expresses the actual mechanism by which complexity degrades performance in an equation relating factors that affect program execution time:

$$\text{Program time} = I \times C \times T$$

where I equals the number of instructions executed for a program, C equals the average number of clock cycles per instruction, and T equals the length of each clock cycle.

The basic RISC principle—one instruction execution per clock cycle with no complex microcoded instructions—obviously makes C smaller. It was by reducing this factor that the first Berkeley RISC chips performed so well against commercial microprocessors. Microcoded machines may take 5 to 10 clock cycles per instruction, versus 1.2 to 1.5 for RISC architectures, Patterson says.

Birnbaum and Stanford's Hennessey contend further that the complex instructions will affect cycle time itself. Birnbaum says the key point is "if you have more complexity and more checking, invariably your cycle time will increase. The machine will pay something in performance for all of the extra combinations . . . there will be one extra step of the decoder, or the error recovery path will be longer. You pay for complexity even when you are not using it." HP's Spectrum design team paid close attention to the three factors expressed by Patterson's equation (see "How HP made architectural tradeoffs on Spectrum," p. 31).

Both HP and Mips Computer Systems made extensive studies of the effect of adding an instruction on a machine's basic cycle time. Loads, stores, and branches account for more than 80% of all instructions, Hennessey says. "It's then very hard to find an instruction that won't hurt the clock speed." The Mips Computer Systems tests paralleled HP's in pinpointing the number of logic levels required, and the way the system deals with exceptions and interrupts, as the main factors tending to increase cycle time.

Because of those tests, says Birnbaum, "whenever someone suggested we really ought to have a wonderful instruction, like 'test left, shift mask, dim the lights,' we had to ask: 'How often will we execute it, and what is the performance degradation?' If you have a hundred instructions, and one of them is executed only 1% of the time, you had better not incur more than a 1% penalty. Our rule was nothing gets in until we do that analysis."

## A THORNY QUESTION

The other factor in the program-execution equation is path length (the number of instructions executed in running each program). It is one of the thornier questions for RISC proponents. Hennessey says a RISC machine will pay about a 30% penalty in added instructions over a machine that uses microcode. "We are willing to take a 30% hit in return for a fivefold improvement in cycles per instruction," he says.

The design goals of the system determine which tradeoffs are acceptable, Birnbaum says. When HP undertook the Spectrum design, one of its goals was to maintain compatibility among the new system and all old machines it would eventually replace. "If we had had no compatibility objective, our job would have been immeasurably simpler," he notes.

As if application-code compatibility was not challenging enough, HP required much more than that. "There's another part—peripheral subsystems, interrupt responses, and input/output compatibility," Birnbaum says. "We worked hard on I/O architecture. The first version HP released in February is essentially our old architecture, because we wanted to use existing peripherals and channel controllers. But what is coming is RISC I/O: direct attachment to the bus from any peripheral. RISC/CISC is down in the noise compared to that problem."

**PRIME MOVER.** John Hennessey spawned a company and a new RISC computer.

As difficult as the RISC choices were, Birnbaum says, they paid off in a great deal more than just simple system performance. "There is a tremendous flexibility that comes from imposing the discipline of not adding anything unless it pays its way," he says. "Building a microprocessor on a smaller chip, as RISC architecture permits, gives you two different ways to play the game. You can make a very small chip and get high yield and lower manufacturing cost. That might be a good idea for a semiconductor manufacturer. But the chip cost isn't terribly significant compared to system cost, so instead we have the option to put more on a single chip. It's easier to put cache memory, or floating-point circuitry, or virtual memory management on the chip, and run at chip speed instead of package speed. The 950, HP's first RISC minicomputer, is a 7-mips machine not because we have 1-$\mu$m technology but because the architecture lets us put the whole processor on a single chip."

Birnbaum predicts that even greater benefits are possible in the future. On-chip coprocessors that straddle the cache and memory buses will make possible very fast special-purpose signal processing, graphics, or encryption systems. "We now have the silicon area to put this circuitry on," he says.

RISC, says Birnbaum, is another iteration of the experimental process in computer design. "In an ideal world, 30 or 40 years from now," he concludes, "people will design computers the way they design airplanes with really good knowledge of aerodynamics and fluid mechanics. But airplanes flew before the science of aerodynamics was invented."

The RISC concept may be at the Kitty Hawk stage, and skeptics may argue that it's just too simple to be practical. But real-world practitioners of computer design retort that the RISC principles are a foundation upon which to build practical machines that are now working well. After all, IBM and HP are no-nonsense outfits—they are not given to taking fliers on shaky propositions. □

## HOW HP MADE ARCHITECTURAL TRADEOFFS ON SPECTRUM

**To William Worley,** principal architect of the Hewlett-Packard Co. Spectrum computer line, the question of reduced-instruction-set computers versus complex-instruction-set computers is not an argument, but a matter of principle.

Worley says three factors affect time to execute a program: average number of clock cycles per instruction, cycle time, and path length (number of instructions used). "RISC reduces the value of the first two of those terms," he says. But making tradeoffs among these three principles is not always straightforward, cautions Worley, who is head of system architecture for the Spectrum project in Cupertino, Calif. Instructions that increase cycle time must be justified by reductions in the other two terms.

The principal influences on cycle time are the way a system handles interrupts and exceptions and the number of logic levels an instruction requires. Interrupts normally affect the critical path of an execution and lengthen it gradually.

The number of logic levels is a step function. "If an instruction requires computation and suboperands that is used in the next instruction, we may reach a number of logic levels that is larger than can be computed in one cycle, and the basic cycle would get larger," Worley explains. "If the instruction repertoire requires $n$ levels of logic to accomplish all of the instructions, an additional instruction requiring $1.5n$ levels would clobber the cycle time."

Data operations can be plotted, since data moves from a register through an arithmetic logic unit and back, Worley says. HP found that the optimum number of logic levels for primitive operations in the instruction set was equal to one more than the minimum possible number of levels.

Instructions requiring more logic levels were broken up. One 8-byte instruction for 32-bit address displacements, for instance, was divided into two 4-byte instructions. That's not the same as resorting to a complex instruction, since it is still executed in direct hardware and does not require microcode, Worley says.

**WILLIAM WORLEY:** Opting for a RISC architecture is a matter of principle.

Simple instructions can reduce cycle time; but one of the basic RISC/CISC arguments is over how to make the tradeoff between many simple instructions and a few complex ones. "We have to learn to keep the path length small as well," Worley says.

One way HP has approached this problem is through the design of the instruction set. In 33 cases, Worley says, his group found ways to combine in a single instruction operations that formerly took two or more. One example: an instruction to compare two quantities, which previously required a comparison and a conditional branch. HP put the operation in one branch.

"Our tradeoffs don't hit path length or cycles per instruction," Worley says. The tradeoffs made in architecture don't take place in a vacuum and must stand up in different markets, he emphasizes. "We have to distinguish between architecture and implementation. Some implementations will realize all of the theoretical efficiencies of RISC. Not all will. Each implementation needs to make sense as a business proposition. We can push the architecture to the limit, but not every time." *—Clifford Barney*

# What's hot and what's not in workstations

CASE and electronic publishing are joining CAD/CAE as the high-growth segments of the U.S. workstation market

By David Card

Engineering workstations became a real live billion-dollar business last year. The fastest growing computer market held up in the face of an industry slump, and should continue to cruise along in 1987. High-growth industries tend to have common trends, and this one will be no exception. Prices will drop and market share will consolidate in the hands of a handful of players, while new applications, such as publishing and software engineering, will solidify.

Sales should reach $1.86 billion in 1987, forecasts market research firm International Data Corp. in Framingham, Mass. IDC analyst Vicki Brown says that unit sales are way up, but that the average selling price of a 32-bit, graphics-oriented, networked single-user computer (IDC's definition of a workstation) shriveled from $30,000 in 1985 to $22,000 last year.

Two key markets emerging this year are electronic publishing and software writing — usually referred to as CASE, for computer-aided software engineering. These applications do not need the high-performance color graphics required by design automation, which is still by far the biggest segment of the workstation market.

One of the big surprises of 1986, IDC's Brown also says, was the emergence of Hewlett-Packard Co. as the workstation leader, although other analysts, including Dataquest Inc. of San Jose, Calif., say Apollo Computer Inc. of Chelmsford, Mass., and Sun Microsystems Inc. of Mountain View, Calif., still dominate the market.

HP, which does not break out its workstation sales but agrees with IDC's conclusions, says its strength was not in selling to design engineering, the traditional destination for workstations. Instead, HP found new homes for workstations, on the factory floor and in the lab, as controllers for clusters of HP instruments. Measurement automation "is becoming a smaller proportion of our sales as other markets like design automation and software engineering come up," reports William Parzybok, vice president and general manager of HP's Engineering Systems Group in Fort Collins, Colo.

## CASEs in point

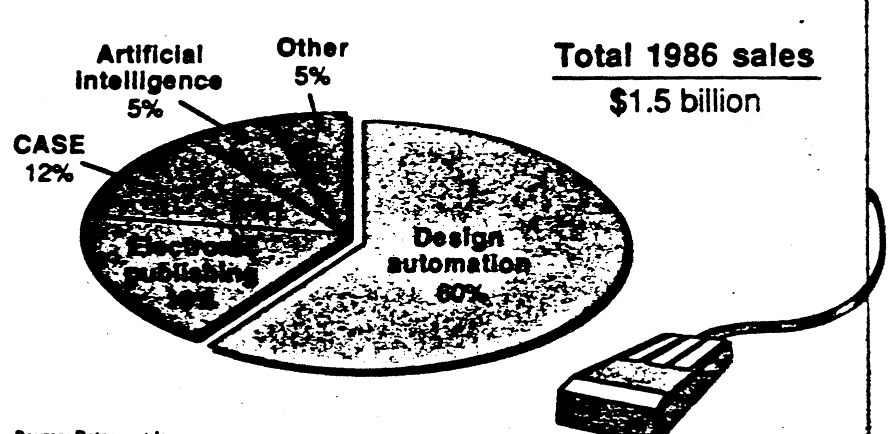
For some analysts, however, including Dataquest, Sun is the undisputed leader in electronic publishing and CASE. Sun president Scott McNealy says that he sells CASE stations to what he calls "internal OEMs," rather than through regular original equipment manufacturers. "More often than not it's an EDS or a Boeing or a Lockheed or a Hughes or somebody like that who's got thousands of software engineers," he says.

"The other area is AI," McNealy says, referring to artificial intelligence. "It's not necessarily a market, but a technology that's probably going to drag us into a lot of weird markets that we wouldn't have gotten into."

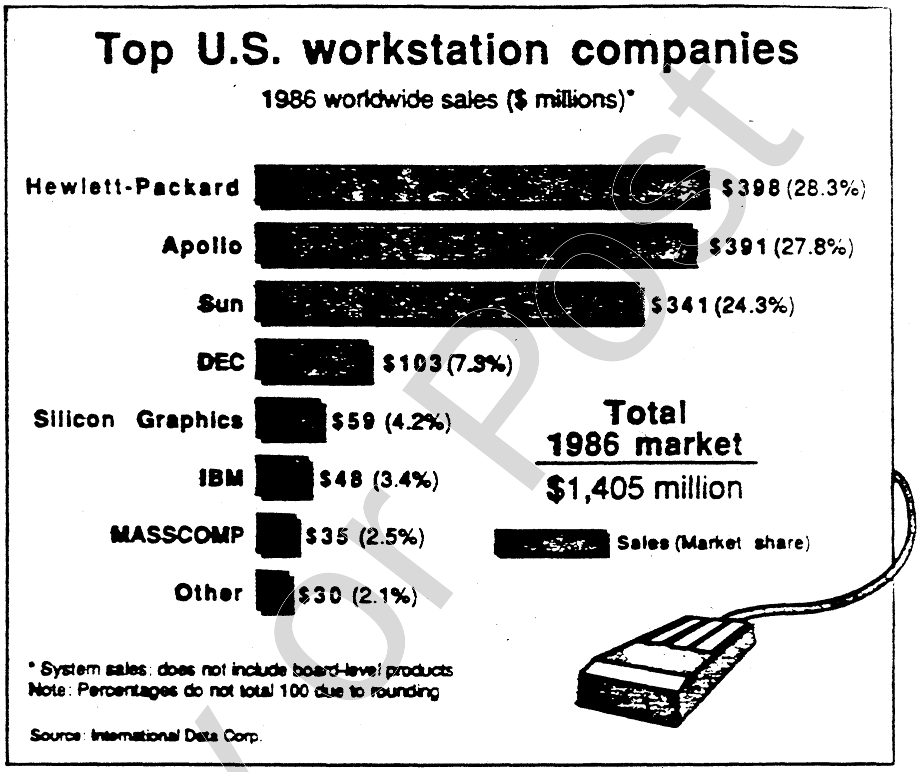
Not surprisingly, McNealy is more excited about the design markets than Sun's current strongholds. Some of those internal OEMs have been using their Suns to write software for computer-aided design, manufacturing and engineering (CAD/CAM/CAE). The CAE and CAD markets have been a little sluggish lately, but will bounce back, McNealy says. "It comes in cycles, and maybe we're down at the bottom, but I see a lot of good development going on," McNealy says. "I haven't given up on it at all."

**Design automation is still the major market for workstations**

Workstation sales by end market




Source: Dataquest Inc.




Anxious to add Sun to its list of original equipment manufacturing (OEM) suppliers is CAE/CAD vendor Racal-Redac Ltd. of Tewkesbury, England. The firm just signed a $25 million extension to its deal with Apollo. Redac also sells systems based on machines from IBM and Digital Equipment Corp. But Redac chief executive Ian Orrock likewise sees Sun as a powerful contender in Europe, Japan and Scandinavia, due to its OEM arrangements with major computer vendors in those regions. "Sun's greatest strength," Orrock adds, "is its software tools."

Apollo, however, late last year struck a blow into Sun's CASE heartland, landing a contract for some 130 workstations for the Software Productivity Consortium. The Consortium is a group of 14 defense contractors — including Allied-Signal Inc., Martin Marietta Corp., TRW Inc. and United Technologies Corp. — founded in September 1985 to streamline and regulate the process of writing software for military systems.

"Since it was not a government procurement, we were free to write a sensible RFP [request for proposal]," quips John Githens, vice president for the Consortium's Environmental Engineering division in Reston, Va. Githens likes Apollo because "they were not just a workstation and/or network vendor." He particularly likes the performance of Apollo's proprietary Domain network. But the Consortium would not have gone with Apollo if the company had not "opened up their network," with ties into networks and hardware from other suppliers.

He says that Apollo was "more forward-looking" and — surprise — more aggressive on pricing than Sun, usually regarded as the low-end leader. Githens says DEC also bid on the contract.

**Publishing for dollars**

Electronic publishing is also proving to be a boom market for workstations, accounting for 18% of 1986's sales, or $270 million, according to Dataquest. Two years ago, that share was only 13% for a $55 million total. Start-ups and established companies alike are rushing to offer software for the layout and design of publications, both for internal consumption and distribution [ELECTRONIC BUSINESS February 15, 1987]. Xerox Corp. was big enough to offer its own workstation as a publishing engine, with its proprietary microprocessor and operating system. But most workstation companies have rallied around the 68000 series of chips from Motorola Inc.

The rise of electronic publishing, meanwhile, has opened up new opportunities for third-party software companies and system integrators. Interleaf Inc. of Cambridge, Mass., for example, offers software for Apollo, DEC and IBM workstations as well as Sun. "Our decision proved right in hindsight," Interleaf marketing vice president George Potter says of opting out of the proprietary hardware business. "Apollo ships in a week what we ship in a year," he says, for example. This way Interleaf can concentrate on software development. At the same time, Interleaf competitor Texet Corp. of Arlington, Mass., last year switched from homemade hardware to Sun, says president Al Ireton, just in time to land a deal with typesetter supplier Compugraphic Corp., which was working on its own Sun-based system.

**Another new market for workstations is financial analysis**

Another new market for workstations, one with "the potential to be the biggest market in this industry," according to IDC's Brown, is financial analysis. "One of our largest single orders for workstations came from a non-engineering market, namely a large banking institution," says Richard Lewan, a marketing manager at DEC in Marlboro, Mass. Lewan says that the computing requirements of financial analysts — number-crunching on the desktop, multiple windows onscreen, tight networking — duplicate those of engineers. "You talk about these people moving billions of dollars around," he says. "They're not afraid to spend some money on a workstation."

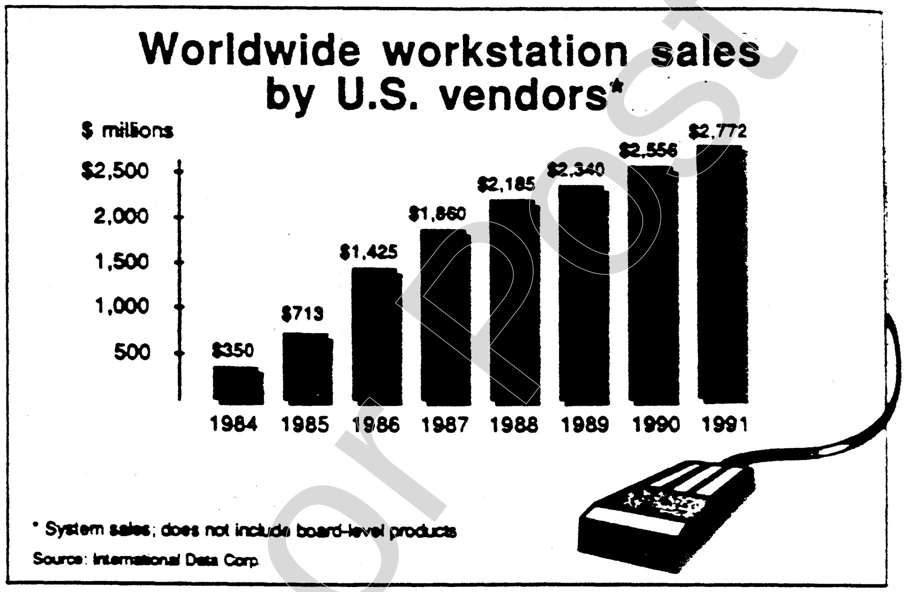
**The players**

As the market matures, the hardware from the players in the workstation arena is getting more and more difficult to differentiate at a casual glance, according to Dataquest analyst David Burdick. Apollo used to be "The Proprietary One" and Sun, with its emphasis on the Unix operating system and Ethernet network, "The Standards One." Now, Apollo offers its version of Unix. HP always did. Standardization will not be a sales pitch in 1987 "because everybody's got it," predicts Edward Zander, Apollo's vice president for corporate marketing.

All three firms — Apollo, HP and Sun — use the 68020 chip from Motorola, but Parzybok hints that it would not be unreasonable to expect Hewlett-Packard to adopt its so-called Precision Architecture — based on proprietary reduced instruction set computing (RISC) chips. The 68030 is due early this summer, Burdick says, and its ready-made 68000 series software base will discourage the workstation companies from looking else-where. The Intel Corp. 80386 is starting to ship, but the three leaders all claim disinterest.

Burdick of Dataquest also predicts that, as the machines become generic, marketing will be the distinguishing factor. Salesmanship will become increasingly important, he says.

Among the crowd, Silicon Graphics Inc. of Mountain View is "a company to look out for," according to Burdick. Silicon Graphics has staked out a cozy niche for itself at the highest end of the workstation market. The company sells 3-D graphics machines running $40,000 to $60,000 into markets such as mechanical CAE, animation and visual simulation. "At the high end, they are starting to be considered a mainstream application and architecture, and they are certainly recognized as the leader there," Burdick contends.




**Worldwide workstation sales by U.S. vendors***

* System sales; does not include board-level products

Source: International Data Corp

Dana Computers Inc. and Stellar Computer Inc. are two start-ups whose principals include famous names from Convergent Technologies Inc. and Apollo, respectively. Both firms profess to deliver by the end of the year ultra-high end graphics engines. Silicon Graphics director for product marketing Craig Olson says he thinks these two could price themselves out of his market.

"I don't see how they could possibly catch Silicon Graphics," IDC's Brown says. Olson adds that the "three key players," to his mind Apollo, DEC and Sun, are not competing in his market yet. His firm currently has an OEM deal with Control Data Corp.

Another new contender in the workstation world is MASSCOMP Corp. of Westford, Mass. "The primary difference between us and most of the others is that we supply product to people that sell workstations," according to Douglas Rowan, vice president for domestic sales and marketing. Harris Corp., and Unisys Corp.'s Graftek subsidiary, for instance. This was not always the case. Before MASSCOMP located its niche in real-time computing and data acquisition, it lost money trying to compete in the general purpose market.

In his business, Rowan says, he runs into DEC and HP as competitors. Right now, MASSCOMP is teaming up with Sun in competition for a 4,000-workstation contract to the National Security Agency, the U.S. hush-hush communications spy agency. Systems Development Corp., a Unisys systems integrator, is bidding the pair against Apollo, and another integrator is bidding IBM hardware, according to Rowan. He says that the NSA spec'd a two-part buy, with the general-purpose machine based on Apollo, and the real-time machine based on MASSCOMP. The winner is expected to be announced this month.

**Four more to watch**

Intergraph Corp., the Huntsville, Ala., CAD/CAM company, is known for picking smart hardware strategies — Intergraph was an early DEC OEM. So president James Meadlock did not exactly shock anyone when he steered Intergraph away from hosts to a distributed Unix workstation-based setup. But he did when he opted to build his own, especially with the

> Smart money would not count either DEC or IBM out of the race

Fairchild Semiconductor Corp.'s Clipper chip as its heart. "It doesn't make sense to run Unix on a DEC workstation," is Meadlock's reasoning.

Now Intergraph is upping the voltage of its shock, with its January announcement that it is plunging wholeheartedly into the general-purpose workstation business. All of this mere months after denying any interest in the OEM market because of the "friction point" between OEMs and the systems suppliers that buy from them. Up until now, workstation sales, bundled with software, have been a little disappointing, Intergraph admits.

Prime Computer Inc. in Natick, Mass., is another CAD/CAM company, at least for a big part of its minicomputer business. Prime promised a workstation a year ago, but so far has not delivered. "It's got to be really hurting them," Brown says. Another minimaker, Data General Corp. of Westborough, Mass., has a strong CAD/CAM presence. "Compared to Apollo and Sun and even Digital, we're a little later getting into this market," admits director of product marketing John Scanlon. "But we have a strong customer base to build on."

The Macintosh personal computer from Apple Computer Inc. of Cupertino, Calif., is not a workstation by any means. However, the rumored second generation Mac will almost certainly contain the same Motorola chip powering most of the machines. Apple's vice president for new business developments, John Scull, will not comment on the subject, nor whether Apple will target workstation markets. But he does say "there is absolutely no question that the power of a workstation will move to the desktop."

Meanwhile, CAE market leader Mentor Graphics Corp. sells systems based on Apollo computers. Vice president Gerald Langeler is keeping a weather eye on Apple from his base in Beaverton, Ore. "It's going to be a very capable product," he says of the next Mac, "every bit as good as the '386 PC. And it may have some goodies that the '386 doesn't have. But it's not clear that they're going to make a big thrust" into CAE.

What's missing from this picture? DEC and IBM, that's what. "Both of these companies did surprisingly poorly," IDC's Brown contends. "The problem DEC and IBM face is, are they going to eat their own children?" Burdick explains — meaning that newer machines could rapidly obsolete their current product line. He says that the two, as leaders in host-based computer systems, are having a hard time managing their way into distributed computing. Or even deciding if that is what they want to do.

In DEC's case, "surprisingly poorly" is a relative term at best. Both analysts count only complete workstations, that is, a computer with graphics. DEC has only offered a color graphics workstation for a year. It has sold at least 25,000 of its MicroVAX II board-level products, Burdick estimates. But he does not count those sales as workstations, because he says most customers are using the MicroVAX II as a three- or four-person mini-host. DEC fans will point out that any dollar spent on a MicroVAX II will not end up in the pockets of Apollo, HP or Sun.

Apollo admits as much. "In technical markets, DEC is the incumbent," Zander says. Besides a massive software base running under its VMS operating system (the MicroVAX can support it or DEC's version of Unix), Zander says DEC is perceived as a safer buy than the younger companies. "Where DEC will win — and they do win — against us, is in the corporate sell," Zander says.

DEC also has no shortage of OEMs. Nearly everyone who sells something that runs on any one workstation, also sells it on the MicroVAX. Smart money would not count either DEC or IBM out of the race.

**And from the East . . .**

Workstation vendors cannot help but give a glance now and then to the Far East. Nearly every Japanese computer company has announced some kind of a workstation, but few have arrived in the United States. Most follow the Motorola/Unix strategy.

To date, however, NEC Corp. is the only Japanese company to bring a workstation to the United States. NEC, which had been embroiled in a messy microprocessor patent infringement suit, stuck to a dual-Motorola chip design for its workstations. "We just wanted to stay in a standard architecture," says Frank Girard, vice president for systems marketing for NEC's American sales arm in Boxborough, Mass.

With a moving video-oriented product, NEC will focus on the automotive and aerospace industries, going after large companies with deep pockets, rather than trying to sign up OEMs, Girard says.

Meanwhile, Japan's Ministry of International Trade and Industry's Sigma project has started to produce workstations as well. And another state-sponsored effort is coming up with a Unix competitor called TRON. That is something way off, Sun's McNealy says. "Let's assume they get it done and they announce it," he says. "It's still going to take another seven or eight years to have it all ready. The industry, practically, is behind Unix. There is not a major company in the industry today that doesn't have Unix in its product line. TRON is going to be a while and there will be time to react." □

# Whither Armonk?

IBM introduced its RT PC — for reduced instruction set computer (RISC) technology — in January 1986 with, for Armonk, no small fanfare. The RT, IBM's workstation, has so far proved something of a flop.

"It has clearly been a very, very big disappointment," says analyst David Burdick of Dataquest Inc., San Jose, Calif. He estimates only 3,000 to 5,000 were sold last year, concluding that the figure is closer to 3,000. "In technical markets you can't get too cute with not having leading-edge performance," Burdick contends. "It has a third to a fifth of the average floating point performance, no networking and clumsy graphics."

"It is not a technological barnburner," agrees Vicki Brown, an analyst with International Data Corp. (IDC) in Framingham, Mass. She expects IBM will correct most of the RT's performance deficiencies in early 1987. Still, without the software or original equipment manufacturers (OEMs) many expected to dog the RT, it will be 1988 before IBM has any significant effect on the market, Brown says.

Valid Logic Systems Inc., a San Jose computer-aided engineering (CAE) company, was one of the first to announce its intent to support the RT. Valid still has not offered an RT-based system. Marketing vice president David Foster runs some software on a proprietary workstation and some on Digital Equipment Corp.'s VAXstation.

"Once CV [Computervision] and others had a move to consider the RT. That has really fallen off," admits Bardwell Salmon, marketing vice president for Computervision Corp. in Bedford, Mass. CV is betting on a Sun workstation to power most of its computer-aided design/manufacturing (CAD/CAM) software. Likewise, Robert Lodi, marketing vice president for VIA Systems Inc. of Chelmsford, Mass., which sells CAD software for integrated circuit design on computers from Apollo, DEC and Sun is unimpressed. "The RT has no advantages in the engineering business," Lodi claims.

Apollo's Zander says that the RT's effect on his business has been negligible. "What a yawn. Everybody always leads to project doomsday scenarios from product announcements," he says. IBM, he thinks, may be "forgetting about our little dinky part of the business. What is it, even if it grows to $3 [billion], $4 billion, compared to the overall computer industry?"

Even though IBM may have sold more RTs than some of its other competitors suspect, DEC's Lewan says he's not worried either. Lewan claims that his customers have a performance "checklist" with minimum requirements for graphics, networking, power and price — none of which the RT meets.

"This will be the year to tell whether IBM will be a strong contender with the RT — whether it will be the RT or the '386," says Brown of IDC. She says that any potential workstation based on Intel Corp.'s 80386 chip would come out of a different division within IBM. The two would end up competitors. "It's not out of the question that IBM would drop it," she says of the losing product. Dataquest's Burdick agrees that this is the year to ask, "Is the RT part of the long-term equation?" He points out that at the end of the RT introduction last year, IBM quietly announced the discontinuation of its 9000 series.

IBM declined to be interviewed for this story. □